

19.683/115/14/2004



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK
PENCARIAN DATA CITRA PADA BASISDATA CITRA
DENGAN MEMANFAATKAN FUNGSI DEKOMPOSISI
WAVELET**

TUGAS AKHIR



Rif
005.1
Kha
P-2
1998

Oleh :

JHANZEIB KHAN

2694 100 042

**PERPUSTAKAAN
ITS**

Tgl. Terima	16-7-2003
Terima Dari	H
No. Agenda Fcp.	210379

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1998**

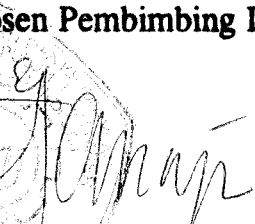
**PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK
PENCARIAN DATA CITRA PADA BASISDATA CITRA
DENGAN MEMANFAATKAN FUNGSI DEKOMPOSISI
WAVELET**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagai Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

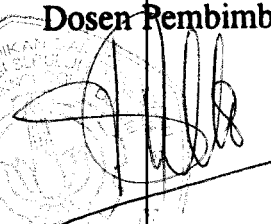
Mengetahui / Menyetujui

Dosen Pembimbing I



Ir. Esther Hanaya, M.Sc.
NIP. 130 816 212

Dosen Pembimbing II



Rully Soelaiman, S.Kom.
NIP. 132 085 802

**SURABAYA
JULI, 1998**

ABSTRAK

Pada proses pencarian citra terhadap sebuah basisdata citra , kendala utama yang biasanya dihadapi adalah lamanya waktu proses pencarian yang dibutuhkan. Hal ini terjadi mengingat proses pencarian dilakukan dengan membandingkan secara langsung antara citra masukan (Image query) dengan seluruh pustaka citra yang terdapat pada basisdata citra sehingga didapatkan citra-citra yang bersesuaian. Persoalan ini terus bertambah dengan terus berkembangnya ukuran dari basisdata yang ada.

Karena itu perlu dipikirkan suatu metode pencarian yang memungkinkan proses pencarian dapat dilakukan dengan cepat, efisien, namun tetap akurat. Dengan memanfaatkan fungsi dekomposisi wavelet multiresolusi dapat dikembangkan suatu perangkat lunak yang memungkinkan proses pencarian dapat dilakukan dengan cepat bahkan terhadap basisdata citra yang berukuran besar sekalipun. Hal ini mungkin dilakukan karena proses pencarian tidak dilakukan langsung terhadap basisdata citra, tetapi terhadap koefisien-koefisien yang mewakili citra-citra, baik citra masukan maupun pustaka citra yang terdapat pada basisdata citra. Koefisien-koefisien tersebut didapat dengan melakukan transformasi wavelet terhadap citra –citra yang terlibat dalam proses pencarian. Format citra yang digunakan dalam perangkat lunak yang dibuat adalah format citra BMP.

KATA PENGANTAR

Segala puji syukur saya panjatkan kehadirat Alloh S.W.T yang telah memberikan rahmat, hidayah, dan kasih sayang yang tak putus-putusnya sehingga saya dapat menyelesaikan Tugas Akhir ini yang berjudul :

**PERANCANGAN DAN PEMBUATAN PERANGKAT
LUNAK Pencarian Data Citra Terhadap
BASISDATA CITRA DENGAN MEMANFAATKAN
FUNGSI DEKOMPOSISI WAVELET**

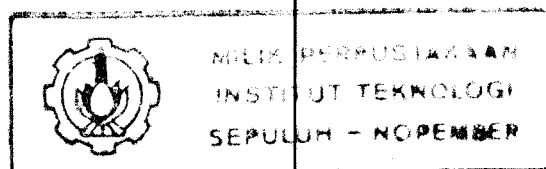
Tugas Akhir yang merupakan syarat akademis bagi para mahasiswa dalam rangka menyelesaikan kuliahnya guna mendapatkan gelar sarjana teknik informatika. Tugas akhir ini memiliki bobot 6 SKS (satuan kredit semester). Tugas akhir ini diberikan kepada mahasiswa untuk memberikan bekal pengalaman dalam hal merancang, membuat, dan mengimplementasikan perangkat lunak sebagai penerapan ilmu-ilmu yang telah dipelajari di jurusan Teknik Informatika.

Dengan terselesaikannya Tugas Akhir ini penulis ingin memberikan penghargaan sebesar-besarnya kepada segala pihak yang telah dengan ikhlas memberikan bantuannya dengan mengucapkan terima kasih yang setulus-tulusnya dan doa semoga Alloh membalas segala bantuan mereka, terutama sekali kepada :

1. Dr. Ir. Arief Djunaidy, M.sc., selaku Ketua Jurusan Teknik Informatika , FTI ITS, yang telah memberikan bimbingan, petunjuk, dan pengarahan selama penulis menempuh studi di Jurusan Teknik Informatika, FTI ITS .

2. Ir . Esther Hanaya, M.sc., selaku dosen pembimbing I yang telah memberikan bimbingan, petunjuk, dan pengarahan kepada penulis dalam upaya menyelesaikan Tugas Akhir ini.
3. Rully Soelaiman, S.Kom., selaku dosen pembimbing II yang juga telah memberikan bimbingan, petunjuk, dan pengarahan kepada penulis dalam upaya menyelesaikan Tuga Akhir ini.
4. Lalaji dan Mami (*all for u mam*) tercinta beserta Nani Ana, Bebe Nur, Ji Duri, Lala Khan, Eva, Bobby, Devi, dan Isykar yang telah dengan rela menjadi penyampai Rahmat Alloh kepada penulis beserta doa yang tak putus-putusnya.
5. Yuyun "*My Jasmine*" atas segala cinta dan semangatnya.
6. Seluruh staf dosen TC atas ilmu dan bimbingan yang diberikan selama masa kuliah, dan seluruh karyawan atas pelayanan manisnya. Semoga Alloh membalas segalanya.
7. Mami "*kost*" Suhari, Ti yok (ojek pribadiku), dan rekan-rekan TC Waskito,Anton, Firdaus atas printernya , Adek, Up pi' keriting "*elek*",Deni, Bilqis, Eko "*soto dan ojeknya*", seluruh rekan-rekan AJK/C0A lainnya "*you are my families for good*",dan seluruh rekan TC lainnya, atas apa saja yang telah penulis terima.
8. Seluruh manusia yang pernah berdoa untuk kebaikan penulis, semoga Alloh membalas segalanya.

Walaupun penulis telah berupaya dengan sebaik-baiknya namun sebagai manusia biasa penulis sadar bahwa apa yang telah tertuang dalam buku ini masih



jauh dari sempurna. Akhir kata penulis penulis berharap semoga Allah S.W.T., yang Maha Indah melimpahkan rahmat, hidayah, serta Inayahnya kepada semua pihak yang tersebut di atas, bangsa dan negara serta seluruh umat manusia. Semoga Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya khususnya bagi rekan-rekan mahasiswa dan masyarakat pada umumnya, serta dapat dikembangkan lebih lanjut guna memenuhi tuntutan kebutuhan dan perkembangan teknologi di masa yang akan datang.

Surabaya, Juli 1998

Jhanzeib Khan

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
 BAB I. PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Sasaran	3
1.4 Pembatasan Masalah	4
1.5 Metodologi	5
1.6 Sistematika Penulisan	5
 BAB II. TRANSFORMASI WAVELET dan ANALISA MULTIRESOLUSI	 7
2.1 Konsep Dasar Aljabar Linier	7
2.1.1 Ruang Vektor	7
2.1.2 Basis dan Dimensi	8
2.1.3 <i>Inner Product</i> dan Orthogonalitas	9
2.1.4 Norm dan Normalisasi	10
2.2 Transformasi Wavelet	10
2.2.1 Jenis – Jenis Transformasi Wavelet	12
2.2.1.1 Transformasi Wavelet Kontinyu	12
2.2.1.2 Transformasi Wavelet Diskrit <i>Redundant</i>	12
2.2.1.3 Transformasi Wavelet Basis Orthogonal	13
2.3 Analisa Multiresolusi	13
2.3.1 Analisa Multiresolusi Pada Ruang $L^2(R)$	13
2.3.2 Representasi Wavelet Pada Citra Dua Dimensi	16

BAB III. PENCARIAN CITRA DENGAN METODE DEKOMPOSISI WAVELET	19
3.1 Pendahuluan	19
3.2 Pendekatan Multiresolusi	19
3.3 Pencarian Citra Dengan Metode Dekomposisi Wavelet	23
BAB IV. PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK	26
4.1 Pendahuluan	26
4.2 Sistem Perangkat Lunak dan Tujuannya	26
4.3 Perancangan dan Implementasi Perangkat Lunak	27
4.3.1 Data Flow Diagram	28
4.3.2 Perancangan Form dan Datanya	31
4.3.2.1 Form Utama	31
4.3.2.2 Form Image Viewer	40
4.3.2.3 Form Set Basisdata	43
4.3.2.4 Form Tambah Data Citra Basisdata	45
BAB V. UJI COBA DAN PEMBAHASANNYA	49
5.1 Uji coba dan Analisisnya	49
5.1.1 Uji Coba dengan Perubahan Pada Citra Masukan	48
5.1.2 Analisis Pengaruh Perubahan Citra Masukan	57
5.1.3 Uji Coba Untuk Mengetahui Tingkat Keberhasilan ..	57
5.2 Analisis Kecepatan Pencarian	61
BAB VI. PENUTUP	63
6.1 Kesimpulan	63
6.2 Saran	64
DAFTAR PUSTAKA	65

DAFTAR GAMBAR

	Halaman
1. Gambar 2.1. Fungsi Skala Haar	15
2. Gambar 3.1. Proses Pencarian Citra Dengan Dekomposisi Wavelet ..	24
3. Gambar 4.1. <i>Data Flow Diagram</i> tingkat 0	29
4. Gambar 4.2. <i>Data Flow Diagram</i> tingkat 1	29
5. Gambar 4.3. <i>Data Flow Diagram</i> tingkat 2 Pra Proses.....	30
6. Gambar 4.4. <i>Data Flow Diagram</i> tingkat 2 Proses Pencarian	31
7. Gambar 4.5. <i>Form</i> Utama	32
8. Gambar 4.6. <i>Form Image Viewer</i>	41
9. Gambar 4.7. <i>Form Setting</i> Basisdata	43
10. Gambar 4.8. <i>Form</i> Tambah Data	46
11. Gambar 5.1. Citra Masukan Pada Percobaan Pertama	50
12. Gambar 5.2. Citra-citra Keluaran Pada Percobaan Pertama	50
13. Gambar 5.3. Citra-citra Keluaran Pada Percobaan Kedua	51
14. Gambar 5.4. Citra-citra Keluaran Pada Percobaan Ketiga	52
15. Gambar 5.5. Citra Masukan Pada Percobaan Keempat	52
16. Gambar 5.6. Citra-citra Keluaran Pada Percobaan Keempat	53
17. Gambar 5.7. Citra-citra Keluaran Pada Percobaan Kelima	54
18. Gambar 5.8. Citra Masukan Pada Percobaan Keenam	54
19. Gambar 5.9. Citra-citra Keluaran Pada Percobaan Keenam.....	52
20. Gambar 5.10. Citra Masukan Pada Percobaan Ketujuh	56

21.	Gambar 5.11. Citra-citra Keluaran Pada Percobaan Ketujuh	56
22.	Gambar 5.12. Citra Sketsa Masukan Pada Percobaan Kedelapan	56
23.	Gambar 5.12. Citra-citra Keluaran Pada Percobaan Kedelapan	58



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

DAFTAR TABEL

	Halaman
1. Tabel 3.1 Nilai Faktor Bobot untuk Struktur Warna RGB	22
2. Tabel 5.1 Tabel Pengaruh Perubahan Citra Pada Kebersihan	58
3. Tabel 5.2 Tabel Kecepatan Hasil Pencarian Citra Pada Percobaan ..	62

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan semakin berkembangnya teknologi informasi dan semakin luasnya penyebaran media digital di seluruh dunia, memungkinkan bagi semua orang untuk melakukan akses terhadap citra-citra digital yang tersebar dan biasanya memanfaatkan teknologi internet. Dari waktu ke waktu kecenderungan ini terus berkembang dengan semakin banyak orang yang memanfaatkan keberadaannya. Hal tersebut menyebabkan terus berkembang pulalah ukuran dari basisdata yang harus disediakan.

Di dalam mengakses sebuah citra, tidak jarang dilakukan proses pencarian citra terhadap sebuah basisdata untuk mendapatkan sebuah citra yang diinginkan atau hanya sekedar untuk mengetahui apakah citra yang dimaksud telah berada dalam basisdata kita. Seiring dengan terus berkembangnya ukuran dari basisdata citra, metode tradisional yang biasa digunakan dalam pencarian citra sudah tidak mungkin digunakan mengingat kesulitan dan lamanya waktu pencarian yang dibutuhkan[Char95]. Hal tersebut terjadi karena memang proses perbandingan antara citra masukan dan citra-citra pada basisdata citra (pustaka citra) dilakukan dengan perbandingan langsung antara citra masukan dengan seluruh pustaka citra pada basisdata sehingga waktu yang dibutuhkan relatif lama. Karena itu perlu dipikirkan untuk dikembangkan suatu metode pencarian yang memungkinkan proses pencarian dilakukan dengan relatif cepat namun tetap akurat.

Dalam tugas akhir ini akan diimplementasikan suatu perangkat lunak yang memungkinkan pencarian citra pada basisdata dapat dilakukan dengan cepat namun tetap akurat bahkan terhadap basisdata citra yang berukuran besar sekalipun. Algoritma pencarian yang dibuat dengan memanfaatkan fungsi dekomposisi wavelet multiresolusi. Metode ini memungkinkan pencarian citra dapat dilakukan dengan cepat, efisien, dan akurat.

1.2 Perumusan Masalah

Data citra yang digunakan dalam pengembangan perangkat lunak ini adalah data citra dengan format BMP, baik yang tersimpan dalam basisdata citra ataupun yang berfungsi sebagai citra masukan. Citra yang digunakan sebagai citra masukan merupakan sebuah citra hasil *scan* atau citra hasil gambar (sketsa). Proses dalam perangkat lunak ini terbagi menjadi dua bagian yaitu proses awal (praproses) dan proses perbandingan (pencarian).

Pada proses awal (praproses) dilakukan terhadap citra-citra yang terdapat dalam basisdata citra yaitu dengan melakukan proses transformasi wavelet terhadap citra-citra tersebut dan terhadap citra masukan sebelum pencarian. Dalam hal ini digunakan proses dekomposisi *haar wavelet* standard dua dimensi. Proses ini meliputi proses dekomposisi satu dimensi terhadap setiap baris piksel citra yang selanjutnya diikuti proses dekomposisi satu dimensi terhadap setiap kolom hasilnya. Dekomposisi dilakukan terhadap seluruh piksel citra pada setiap nilai penyusun pikselnya (*red, green, blue*). Setelah proses dekomposisi dilakukan nilai piksel posisi $[0,0]$ akan berisi nilai rerata dari seluruh nilai piksel dalam citra

tersebut dan posisi lainnya berisi nilai-nilai koefisien wavelet. Nilai-nilai ini berguna untuk merekonstruksi nilai citra asli (asal) sebelum dilakukan transformasi. Namun dalam tugas akhir ini hal tersebut tidak dilakukan.

Nilai-nilai koefisien tersebut tidak seluruhnya perlu digunakan, namun hanya dipilih berapa nilai-nilai yang dianggap paling dapat memawalikan keberadaan citranya dan nilai rerata piksel untuk semua struktur warna R/G/B. Pada akhirnya untuk mengoptimalkan program, nilai-nilai koefisien wavelet tidak perlu disimpan, namun hanya nilai rerata, posisi piksel wavelet terpilih dan tanda nilainya (positif / negatif). Penyimpanan dilakukan dengan memanfaatkan sebuah file.

Sebelum proses pencarian dilakukan, dilakukan proses dekomposisi terhadap citra masukan untuk mendapatkan nilai rerata warna dan koefisien-koefisien wavelet yang dianggap paling dapat mewakili keberadaan citranya. Selanjutnya proses pencarian mulai dilakukan dengan proses perbandingan nilai rerata warna dan posisi koefisien-koefisien wavelet terpilih dari citra masukan dengan koefisien-koefisien wavelet terpilih pustaka citra untuk mendapatkan nilai bedanya. Pustaka citra yang memberikan nilai beda hasil perbandingan terkecil terhadap citra masukan dianggap sebagai pustaka citra yang paling bersesuaian dengan citra masukan dan sebagai citra hasil pencarian yang terbaik.

1.3 Tujuan dan Sasaran

Tujuan yang ingin dicapai dari tugas akhir ini meliputi :

1. Mempelajari efektifitas penggunaan fungsi dekomposisi wavelet dalam proses pencarian citra terhadap basisdata citra.
2. Mengimplementasikan metode pencarian citra yang baik dan dapat melakukan pencarian citra dengan perbandingan citra yang cepat namun tetap akurat bahkan terhadap basisdata citra yang berukuran besar sekalipun.

Sasaran yang akan dilakukan adalah merancang dan membuat perangkat lunak yang dapat melakukan proses pencarian citra dengan cepat, efisien, namun tetap akurat.

I.4 Pembatasan Masalah

Dalam tugas akhir ini akan diimplementasikan perangkat lunak pencarian citra dengan memanfaatkan fungsi dekomposisi wavelet multiresolusi dalam bahasa pemrograman C dengan sistem operasi windows.

Citra yang digunakan berupa citra dengan format BMP dan struktur warna R,G,B dengan pembatasan warna hingga 16 juta warna.

Untuk masukan dan keluaran dibatasi sebagai berikut:

1. Citra masukan yang diberikan bisa berupa citra hasil *scan* dengan format BMP atau citra yang berupa sketsa hasil penggambaran langsung pada editor citra yang disediakan.
2. Citra-citra keluaran yang diberikan berupa beberapa buah citra yang memiliki perbedaan terkecil menurut hasil perhitungan terhadap citra masukan.

I.5 Metodologi

Penyelesaian tugas akhir ini dilakukan dengan langkah-langkah sebagai berikut :

1. Studi literatur yang meliputi mempelajari transformasi wavelet, mempelajari algoritma pencarian citra (perbandingan citra), dan mempelajari metode pengurutan yang dibutuhkan (dalam hal ini *heap sort*).
2. Perancangan Sistem dan Struktur Data.
3. Perancangan dan Pembuatan Program.
4. Implementasi Program.
5. Evaluasi dan Revisi Program.
6. Penulisan Naskah Tugas Akhir.

I.6 Sistematika Penulisan

Naskah tugas akhir ini dibagi menjadi enam bab dengan menggunakan sistematika penulisan sebagai berikut :

Bab I. Berisi pendahuluan, yang menjelaskan latar belakang, perumusan masalah, tujuan dan sasaran dari tugas akhir, pembatasan masalah yang dibahas, metodologi, dan yang terakhir menjelaskan sistematika penulisan tugas akhir.

Bab II. Menjelaskan tentang landasan teori transformasi wavelet dan analisa multiresolusi.

Bab III. Berisi tentang landasan teori dari metode pencarian citra.



Bab IV. Menjelaskan tentang perancangan perangkat lunak dan implementasinya.

Bab V. Menjelaskan tentang pembahasan hasil percobaan.

Bab VI. Berisi kesimpulan dan saran.

BAB II

TRANSFORMASI WAVELET dan ANALISA MULTIRESOLUSI

Wavelet merupakan alat bantu matematis yang mampu melakukan dekomposisi terhadap sebuah fungsi secara terhirarki[Eric95]. Wavelet dapat digunakan untuk menggambarkan sebuah model/gambar asli ke dalam fungsi matematis, tanpa memperhatikan bentuk dari model merupakan citra, kurva, atau sebuah bidang. Tranformasi Wavelet merupakan sebuah fungsi konversi yang dapat membagi fungsi atau sinyal ke dalam komponen frekuensi atau skala yang berbeda, dan selanjutnya dapat dipelajari setiap komponen tersebut dengan resolusi tertentu sesuai dengan skalanya. Mengingat konsep dasar dari analisa multiresolusi adalah aljabar linier, maka pembahasan akan diawali dengan menjelaskan secara singkat konsep dasar aljabar linier yang berhubungan dengan analisa multiresolusi.

2.1 Konsep Dasar Aljabar Linier

Notasi-notasi yang dipakai untuk membahas transformasi wavelet adalah sebagai berikut :

- \mathbb{Z} melambangkan himpunan bilangan Integer.
- \mathbb{R} melambangkan himpunan bilangan Real.

2.1.1 Ruang Vektor

Ruang vektor (V) dapat didefinisikan sebagai kumpulan vektor-vektor yang elemen-elemennya memenuhi sifat-sifat sebagai berikut :

- Untuk setiap $a, b \in R$ dan $u, v \in V$ berlaku $au + bv \in V$.
- Ada elemen $0 \in V$ sedemikian hingga :
 - a. Untuk setiap $u \in V$; $0u = 0$ dan
 - b. Untuk setiap $u \in V$; $0 + u = u$.

Setiap elemen dalam ruang vektor disebut vektor dan 0 disebut dengan vektor nol.

Pada pembahasan ini vektor dipandang sebagai sebuah fungsi dan beberapa ruang vektor yang digunakan adalah :

- $L^2(R)$ melambangkan ruang vektor dari fungsi $f(x)$ yang terukur dan terintegral secara kuadrat. Untuk $f(x), g(x) \in L^2(R)$, maka *inner product* dari $f(x)$ dengan $g(x)$ ditulis sebagai :

$$\langle f(x) | g(x) \rangle = \int_{-\infty}^{+\infty} f(x) \overline{g(x)} dx,$$

dengan $\overline{g(x)}$ merupakan kompleks konjugate dari $g(x)$.

- $L^2(R^2)$ adalah ruang vektor dari fungsi $f(x, y)$ yang terukur dan terintegral secara kuadrat. Untuk $f(x, y), g(x, y) \in L^2(R^2)$, *inner product* $f(x, y)$ dan $g(x, y)$ ditulis sebagai :

$$\langle f(x, y) | g(x, y) \rangle = \int_{-\infty}^{+\infty} f(x, y) \overline{g(x, y)} dx dy,$$

2.1.2 Basis dan Dimensi

Himpunan vektor $\{u_1, u_2, \dots\}$ dikatakan *linier independence* jika,

$$c_1 u_1 + c_2 u_2 + \dots = 0 \Leftrightarrow c_1 = c_2 = \dots = 0$$

Himpunan vektor $\{u_1, u_2, \dots\} \in V$ dari vektor-vektor *linier independence* merupakan basis dari V jika setiap $v \in V$ dapat dituliskan sebagai :

$$v = \sum_i c_i u_i$$

dengan $c_1, c_2, \dots \in R$. Vektor-vektor dalam sebuah basis untuk V dikatakan *span* V . Secara intuitif dapat dikatakan bahwa *linear independence* adalah tidak adanya vektor yang *redundant*, dan sebuah basis terdiri dari *minimal complete set* vektors. Sebuah ruang vektor tak nol V dinamakan berdimensi berhingga (*finite dimensional*) jika ruang vektor tersebut mengandung sebuah himpunan berhingga dari vektor-vektor $\{ u_1, u_2, \dots, u_n \}$ yang membentuk sebuah basis dan dikatakan bahwa V berdimensi n . Jika tidak maka dikatakan bahwa V berdimensi tak berhingga (*infinite dimensional*) [How91]. Sebagai contoh, dalam ruang tiga dimensi ada tiga vektor dalam basis yaitu :

$$e^1 = (1, 0, 0), e^2 = (0, 1, 0), e^3 = (0, 0, 1).$$

2.1.3 Inner product dan Orthogonalitas

Inner product pada sebuah ruang vektor V bernotasi $\langle \cdot | \cdot \rangle$, pada analisa multiresolusi *inner product* standard yang digunakan didefinisikan sebagai :

$$\langle f | g \rangle = \int_{-\infty}^{+\infty} f(x)g(x)dx$$

yang pada sebuah ruang vektor V merupakan sebuah pemetaan :

$$V \times V \rightarrow R$$

dengan sifat-sifat :

- Simetrik, yaitu : $\langle u | v \rangle = \langle v | u \rangle$
- Bilinier , yaitu : $\langle au + bv | w \rangle = a\langle u | w \rangle + b\langle v | w \rangle$ dan
- Definite positif, yaitu : $\langle u | u \rangle > 0$, untuk $\forall u \neq 0$

Salah satu sifat penting dari *inner product* adalah orthogonalitas. Dua vektor u, v dikatakan orthogonal jika $\langle u | v \rangle = 0$. Sebuah himpunan $\{u_1, u_2, \dots\} \in V$ dikatakan *mutually orthogonal* jika himpunan vektor tersebut bersifat *linier independence* [Eric95]. Sedangkan sebuah basis dikatakan orthogonal jika vektor-vektor itu bersifat *mutually orthogonal*.

2.1.4 Norm dan Normalisasi

Norm merupakan sebuah fungsi yang digunakan untuk mengukur panjang dari sebuah vektor. Norm dari $f(x) \in L^2(R)$ didefinisikan sebagai :

$$\|f\|^2 = \int_{-\infty}^{+\infty} |f(x)|^2 dx$$

Sebuah vektor u dikatakan ternormalisasi jika norm dari vektor u sama dengan

$$\text{satu, atau : } \|u\| = 1$$

Sebuah basis orthogonal yang ternormalisasi dalam norm $L^2(R)$ disebut dengan basis orthonormal, atau dengan kata lain basis u_1, u_2, \dots orthonormal jika :

$$\langle u_i | u_j \rangle = \delta_{ij} ; \text{ dengan } \delta_{ij} = \begin{cases} 1; \text{ untuk } i = j \\ 0; \text{ untuk } i \neq j \end{cases}$$

2.2 Tranformasi Wavelet

Tranformasi wavelet merupakan sebuah fungsi konversi yang dapat dipakai untuk membagi suatu fungsi atau sinyal ke dalam komponen frekuensi yang berbeda, dan selanjutnya dapat dipelajari komponen-komponen tersebut sesuai dengan skalanya [Eric96]. Wavelet merupakan sebuah fungsi variabel real t ,

diberi notasi $\Psi(t)$ dalam ruang fungsi $L^2(R)$. Fungsi ini dihasilkan oleh parameter dilatasi dan rotasi, yang dinyatakan dalam persamaan:

$$(2.1) \dots \Psi_{ij} = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right)$$

dimana a adalah parameter dilatasi dan b adalah parameter translasi. Fungsi wavelet dalam persamaan (2.1) diperkenalkan oleh Grossman dan Morlet [Dav94]. Fungsi ψ disebut sebagai *mother wavelet*, dan fungsi ini harus memenuhi kondisi : [Dav92]

$$(2.2) \dots \int_{-\infty}^{+\infty} \psi(t) dt = 0$$

Biasanya dipilih :

$$\psi(t) = (1-t^2) \exp(-t^2/2)$$

yang merupakan turunan kedua dari fungsi Gaussian.

Salah satu contoh dari *mother wavelet* adalah *Haar wavelet*, yang dinyatakan dengan : [Eric95]

$$(2.3) \dots \Psi_{ij} = \Psi(2^j x - i) \text{ untuk } i = 0, \dots, 2^j - 1$$

dengan,

$$\psi(x) = \begin{cases} 1; & 0 \leq x \leq 1/2; \\ -1; & 1/2 < x \leq 1 \\ 0; & \text{lainnya} \end{cases}$$

Fungsi *haar wavelet* ini merupakan fungsi wavelet yang paling sederhana, tetapi untuk aplikasi-aplikasi yang memerlukan fungsi wavelet yang cukup halus maka

fungsi ini tidak dapat digunakan [Eric96]. Penerapan *haar wavelet* ini akan diperjelas pada analisa multiresolusi.

2.2.1 Jenis-Jenis Transformasi Wavelet

Transformasi wavelet dibedakan menjadi dua tipe, yaitu transformasi wavelet kontinyu dan transformasi wavelet diskrit. Transformasi wavelet diskrit masih bisa dibedakan lagi menjadi dua jenis, yaitu transformasi wavelet diskrit *redundant* (berbentuk *frame*) dan transformasi wavelet basis orthonormal. Faktor yang membedakan dua tipe transformasi kontinyu dan diskrit tersebut terletak pada nilai parameter translasi dan dilatasinya. Berikut akan dijelaskan sedikit mengenai karakteristik masing-masing tipe.

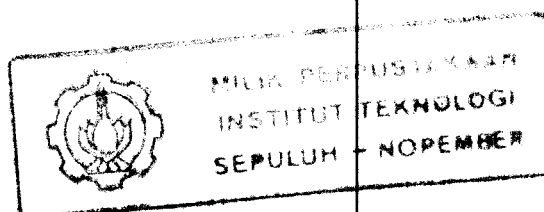
2.2.1.1 Transformasi Wavelet Kontinyu

Transformasi wavelet kontinyu ditentukan oleh nilai parameter dilatasi (a) dan parameter translasi (b) yang bervariasi secara kontinyu, dimana $a, b \in \mathbb{R}$ dan $a \neq 0$. Fungsi translasi wavelets ini dinyatakan dengan :

$$(2.4) \dots\dots\dots (T^{wav} f)(a, b) = |a|^{-1/2} \int (t) \psi\left(\frac{t-b}{a}\right) dt$$

2.2.1.2 Transformasi Wavelet Diskrit *Redundant*

Pada transformasi wavelet diskrit jenis ini, parameter dilatasi dan parameter translasinya dibatasi pada nilai-nilai diskrit. Diskritasi dari parameter-



parameter tersebut adalah: $a = a_0^m$ dan $b = nb_0 a_0^m$. Dimana $m, n \in \mathbb{Z}$ dan ditentukan suatu nilai tertentu untuk $a_0 > 1$ dan $b_0 > 0$, sehingga fungsi waveletnya berbentuk :

$$(2.5) \dots \psi_{m,n}(x) = a_0^{-m/2} \psi(a_0^{-m}(x - nb_0 a_0^m)) \\ = a_0^{-m/2} \psi(a_0^{-m} x - nb_0)$$

Transformasi ini dapat menyebabkan adanya redundansi pada penjabaran fungsi aslinya, hal ini disebabkan perhitungan nilai transformasi waveletnya dilakukan secara pendekatan tanpa perlu menghitung secara tepat.

2.2.1.3 Transformasi Wavelet Basis Orthogonal

Transformasi wavelet ini biasa disebut juga dengan analisa multiresolusi . Dengan memilih nilai-nilai tertentu untuk *mother wavelet* ψ dan a_0, b_0 maka akan didapat nilai $\psi_{m,n}$ yang mempunyai nilai basis orthonormal pada ruang vektor $L^2(\mathbb{R})$. Umumnya dipilih nilai parameter translasi $a_0 = 2$ dan $b_0 = 1$, sehingga fungsi waveletnya dapat berupa :

$$(2.6) \dots \psi_i^j = 2^{-j/2} \psi(2^{-j} x - i)$$

yang dikenal sebagai *haar wavelet*, dimana $\psi(x)$ dipenuhi oleh persamaan (2.3) dan $i, j \in \mathbb{Z}$.

2.3 Analisa Multiresolusi

2.3.1 Analisa Multiresolusi Pada Ruang $L^2(\mathbb{R})$

Analisa multiresolusi pada ruang vektor $L^2(\mathbb{R})$ merupakan konsep penguraian sinyal dalam ruang fungsi $L^2(\mathbb{R})$ ke bentuk multiresolusi dan

didefinisikan sebagai barisan dari subruang tertutup $V_{j \in \mathbb{Z}}$ dari $L^2(\mathbb{R})$ dan dengan pendekatan multiresolusi pada wavelet memungkinkan kita mendapatkan suatu fungsi $\psi(x) \in L^2(\mathbb{R})$ yang memiliki sifat orthonormal terhadap basis ruangnya.

Haar wavelet yang telah dijelaskan di atas merupakan salah satu dari basis yang dapat digunakan dalam analisa multiresolusi. Langkah awal dari analisa multiresolusi adalah himpunan himpunan bersarang dari ruang vektor yang memiliki sifat:

$$V^0 \subset V^1 \subset V^2 \subset \dots$$

resolusi dari fungsi yang terdapat pada ruang vektor V^j akan meningkat pula seiring dengan meningkatnya nilai dari j . Fungsi basis dari ruang vektor V^j biasa disebut dengan fungsi skala yang biasanya dilambangkan dengan ϕ [Eric95]. Kondisi umum dari fungsi skala adalah [Eric95].

$$\int_{-\infty}^{+\infty} \phi(x) dx = 1$$

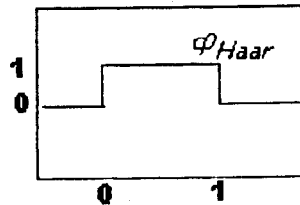
Contoh dari fungsi skala adalah fungsi *box* yang merupakan sebuah fungsi translasi dan dilatasi yang dinyatakan dalam persamaan [Eric95] :

$$(2.7) \dots \phi_i^j(x) = \phi(2^j x - i), i = 0, \dots, 2^j - 1$$

disebut sebagai *box basis*, dimana:

$$\phi(x) = \begin{cases} 1 & \text{; untuk } 0 \leq x < 1 \\ 0 & \text{: lainnya} \end{cases}$$

Fungsi skala pada persamaan (2.7) dilukiskan pada gambar 2.1



Gambar 2.1 : Fungsi Skala dari *haar wavelet* [Dau92]

Bentuk ternormalisasi dari fungsi skala di atas adalah [Eric95]:

$$(2.8) \dots \phi_i^j(x) = 2^{-j/2} \phi(2^{-j}x - i), i = 0, \dots, 2^j - 1$$

Vektor-vektor basis dari fungsi *box* di atas merupakan vektor-vektor basis yang *linier independence* maka vektor-vektor basis tersebut bersifat *mutually orthogonal* atau dalam kata lain fungsi basis di atas merupakan fungsi basis yang orthogonal, dengan demikian bentuk ternormalisasinya pada persamaan (2.8) merupakan basis orthonormal dari ruang vektor V^j .

Langkah selanjutnya dari analisa multiresolusi adalah mendefinisikan ruang-ruang wavelet (*wavelet spaces*). Untuk setiap j didefinisikan W^j ruang vektor baru yang merupakan komplemen orthogonal V^j terhadap V^{j+1} . Dalam kata lain bisa dikatakan bahwa W^j merupakan ruang dari semua fungsi pada V^{j+1} yang orthogonal terhadap semua fungsi pada V^j dengan menggunakan aturan *inner product* yang sama. Dalam hal ini biasanya digunakan *standard inner product* yang didefinisikan :

$$\langle f | g \rangle = \int_{-\infty}^{+\infty} f(x)g(x)dx$$

Fungsi-fungsi bebas linier (*linierly independence*) yang merupakan basis dari W^J ($\psi_i^J(x)$) disebut sebagai *wavelet* [Eric95]. Fungsi-fungsi basis ini memiliki dua karakteristik penting, yakni:

1. Fungsi-fungsi basis ψ_i^J dari ruang W^J , bersama dengan fungsi-fungsi basis ϕ_i^J dari ruang V^J membentuk basis untuk ruang V^{J+1} .
2. Setiap fungsi basis ψ_i^J dari ruang W^J adalah orthogonal kepada setiap fungsi basis ϕ_i^J dari ruang V^J dengan menggunakan *inner product* yang sama.

Wavelet-wavelet yang berhubungan dengan *box basis* di atas dikenal sebagai *haar wavelet* yang terdapat pada persamaan (2.6).

2.3.2 Representasi Wavelet Pada Citra Dua Dimensi

Model wavelet dapat dengan mudah digeneralisasikan ke ukuran dimensi lain dimana $n > 0$ [Osc96]. Untuk memproses citra yang pada umumnya memiliki dimensi dua menuntut model wavelet di atas untuk diturunkan dalam bentuk dua dimensi, sehingga dapat diimplementasikan untuk memproses citra.

Sinyal asli sekarang berbentuk $f(x,y)$ dimana $f(x,y) \in L^2(R^2)$. Pendekatan multiresolusi dari $L^2(R^2)$ adalah barisan dari sub ruang $L^2(R^2)$ yang memenuhi sifat-sifat multiresolusi yang sama dengan sifat-sifat multiresolusi satu dimensi namun dalam hal ini dilakukan secara dua dimensi. Misalkan $(V_j)_{j \in \mathbb{Z}}$ adalah pendekatan multiresolusi dari ruang $L^2(R^2)$. Sinyal pendekatan $f(x,y)$ pada resolusi

j adalah sama dengan proyeksi orthogonal pada ruang vektor V_j . Seperti juga pada fungsi satu dimensi, pada fungsi dua dimensi pun terdapat pula fungsi skala yang unik $\phi(x, y)$ dimana dilatasi dan translasinya memberikan suatu basis orthonormal dari masing-masing ruang V_j . Misalkan

$$\phi_{j,n,m}(x, y) = \phi_{j,n}(x)\phi_{j,m}(y), \text{ maka secara sama fungsi-fungsi}$$

$$(2.9) \dots\dots\dots (2^{-j}\phi(2^{-j}x - n, 2^{-j}y - m)) ; n, m \in \mathbb{Z}.$$

Secara berturut-turut akan membentuk basis orthonormal untuk V_j . Fungsi $\phi(x, y)$ bersifat unik terhadap pendekatan multiresolusi dari ruang $L^2(\mathbb{R}^2)$.

Untuk pendekatan multiresolusi yang dapat dipisah dari $L^2(\mathbb{R}^2)$, maka ruang vektor V_j dapat didekomposisi menjadi :

$$\begin{aligned} V_{j+1} &= V_{j+1} \times V_{j+1} \\ &= (V_j + W_j) \times (V_j + W_j) \\ &= (V_j \times V_j) + (W_j \times V_j) + (V_j \times W_j) + (W_j \times W_j) \\ &= V_j + W_j \end{aligned}$$

Terlihat bahwa W_j terdiri dari tiga bentuk dengan basis orthonormalnya, diberikan oleh :

$$\psi_{j,n}(x)\phi_{j,m}(y) \text{ untuk } W_j \times V_j$$

$$\phi_{j,n}(x)\psi_{j,m}(y) \text{ untuk } V_j \times W_j$$

$$\psi_{j,n}(x)\psi_{j,m}(y) \text{ untuk } W_j \times W_j$$

Hal ini menunjukkan ada tiga buah wavelet, yaitu :

$$\Psi^1(x, y) = \phi(x)\psi(y)$$

$$\Psi^2(x, y) = \psi(x)\phi(y)$$

$$\Psi^3(x, y) = \psi(x)\psi(y)$$

yang merupakan basis orthonormal bagi W_J .

Sedemikian hingga :

$$(2.10) \dots\dots\dots 2^{-j} \Psi^1_{j,n,m}(2^{-j}x - n, 2^{-j}y - m),$$

$$(2.11) \dots\dots\dots 2^{-j} \Psi^2_{j,n,m}(2^{-j}x - n, 2^{-j}y - m), \text{ dan}$$

$$(2.12) \dots\dots\dots 2^{-j} \Psi^3_{j,n,m}(2^{-j}x - n, 2^{-j}y - m).$$

bersama-sama dengan fungsi skalanya merupakan basis orthonormal dari $L^2(\mathbb{R}^2)$

yang merupakan implementasi wavelet secara dua dimensi.

BAB III

PENCARIAN CITRA DENGAN METODE DEKOMPOSISI WAVELET

3.1 Pendahuluan

Permasalahan utama dalam pencarian citra adalah proses perhitungan nilai beda antara citra masukan (Q) dan pustaka citra (T). Nilai ini digunakan sebagai pembanding guna mendapat pustaka citra yang paling bersesuaian dengan nilai beda terkecil. Untuk mendapatkan nilai beda ini digunakan sebuah metrik yang berisikan sebuah nilai yang menggambarkan perbedaan antara citra masukan dengan pustaka citranya. Bentuk paling nyata dari metrik itu adalah [Char95]:

$$(3.1) \dots \dots \dots \|Q, T\| = \sum_{i,j} |Q[i, j] - T[i, j]|$$

atau ,

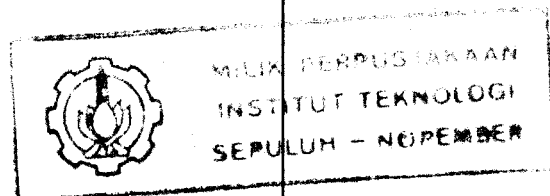
$$(3.2) \dots \dots \dots \|Q, T\| = \left(\sum_{i,j} (Q[i, j] - T[i, j])^2 \right)^{1/2}$$

Persamaan (3.2) merupakan bentuk ternormalisasi dari persamaan (3.1).

Meskipun kedua bentuk metrik diatas merupakan bentuk paling sederhana tetapi sangat mahal untuk dikomputasikan, mengingat proses perbandingan dilakukan dengan membandingkan langsung setiap piksel citra masukan terhadap piksel pustaka citra pada posisi yang bersesuaian. Hal tersebut akan memerlukan waktu pencarian yang sangat lama, terutama jika pencarian dilakukan terhadap basisdata citra yang berukuran besar.

3.2 Pendekatan Multiresolusi

Tujuan utama pendekatan ini adalah untuk mendapatkan suatu bentuk metrik yang relatif cepat dalam komputasi, memanfaatkan penyimpanan yang relatif



--	--

kecil, namun bisa menggambarkan nilai beda secara baik. Pemanfaatan fungsi dekomposisi wavelet dua dimensi terhadap citra dapat dihasilkan sebuah bentuk metrik yang dapat memenuhi kriteria diatas, karena dekomposisi wavelet terhadap sebuah citra memungkinkan dilakukan penaksiran yang cukup baik terhadap keberadaan suatu citra hanya dengan menggunakan sedikit koefisien hasil dekomposisi yang dianggap paling berpengaruh [Cha95]. Pembatasan penggunaan jumlah koefisien hasil dekomposisi ini sangat menguntungkan karena proses pencarian hanya dilakukan dengan membandingkan nilai koefisien hasil dekomposisi terpilih baik dari citra asli maupun dari pustaka citra dengan menggunakan fungsi metrik [Cha95] :

$$[3.3] \dots \dots \dots \|Q, T\| = w_{0,0} |Q[0,0] - T[0,0]| + \sum_{i,j} w_{i,j} |\bar{Q}[i,j] - \bar{T}[i,j]|$$

dimana,

- Q melambangkan citra masukan
- T melambangkan pustaka citra.
- $\bar{Q}[i,j]$ melambangkan nilai koefisien hasil dekomposisi dari citra masukan yang dianggap berpengaruh dan telah dikuantisasi.
- $\bar{T}[i,j]$ melambangkan nilai koefisien hasil dekomposisi dari pustaka citra yang dianggap berpengaruh dan telah dikuantisasi.
- $w[i,j]$ merupakan nilai faktor bobot pada posisi $[i,j]$.

dimana kuantisasi terhadap nilai koefisien dilakukan dengan ketentuan [Cha95]:

$$\bar{Q}[i,j] = \begin{cases} +1, & Q[i,j] > 0 \\ 0, & Q[i,j] = 0 \\ -1, & Q[i,j] < 0 \end{cases}$$

begitu pula terhadap nilai koefisien dari pustaka citra T .

Fungsi (3.3) masih mungkin disederhanakan dengan melakukan beberapa langkah, meliputi [Cha95]:

1. Menggantikan perhitungan beda antara koefisien wavelet $|\bar{Q}[i, j] - \bar{T}[i, j]|$ pada

fungsi metrik (3.3) dengan $(\bar{Q}[i, j] \neq \bar{T}[i, j])$, dimana :

$$(\bar{Q}[i, j] \neq \bar{T}[i, j]) = \begin{cases} 1, & \bar{Q}[i, j] \neq \bar{T}[i, j] \\ 0, & \text{lainnya} \end{cases}$$

perubahan diatas tidak mempengaruhi efektifitas dari persamaan (3.3) tetapi justru akan meningkatkan kecepatan proses komputasi.

2. Dilakukan pengelompokan terhadap nilai faktor bobot $w[i, j]$ sehingga hanya digunakan sedikit nilai faktor bobot yang didapat melalui eksperimen terhadap dekomposisi wavelet terhadap citra. Pengelompokan dilakukan berdasarkan skala dari fungsi waveletnya, dengan memanfaatkan sebuah fungsi sederhana $bin(i, j)$ yang digambarkan sebagai:

$$bin(i, j) = \min\{\max\{i, j\}, 5\}$$

Nilai-nilai faktor bobot yang digunakan untuk struktur warna citra RGB dapat dilihat pada tabel dibawah ini [Cha95]:

Tabel 3.1. Nilai faktor bobot untuk struktur warna RGB [Cha95].

<i>b</i>	<i>Citra Hasil Scan</i>			<i>Citra Sketsa</i>		
	$w^R[b]$	$w^G[b]$	$w^B[b]$	$w^R[b]$	$w^G[b]$	$w^B[b]$
0	5.00	19.21	34.37	4.04	15.14	22.62
1	0.83	1.26	0.36	0.78	0.92	0.40
2	1.01	0.44	0.45	0.46	0.53	0.63
3	0.52	0.53	0.14	0.42	0.26	0.25
4	0.47	0.28	0.18	0.41	0.14	0.15
5	0.30	0.14	0.27	0.32	0.07	0.38

3. Apabila diantara nilai koefisien hasil kuantisasi koefisien wavelet citra masukan terdapat nilai $\overline{Q}[i, j] = 0$, maka nilai itu diabaikan sebab dianggap tidak berpengaruh terhadap proses perbandingan citra.

4. Untuk mendapatkan proses hitungan yang lebih efisien, pada bagian $(\overline{Q}[i, j] \neq \overline{T}[i, j])$ pada persamaan diatas dapat dituliskan sebagai:

$$\sum_{i,j: \overline{Q}[i,j] \neq 0} w_k - \sum_{i,j: \overline{Q}[i,j] \neq 0} w_k (\overline{Q}[i, j] = \overline{T}[i, j])$$

karena bagian $\sum_{i,j: \overline{Q}[i,j]}$ merupakan bagian yang berdiri sendiri tak terpengaruh

oleh keberadaan pustaka citra T (*independence terhadap T*) kita bisa mengabaikannya dengan anggapan tidak mempengaruhi nilai metrik hubungan Q dan T . Langkah ini akan mempengaruhi kecepatan komputasi dengan

asumsi bahwa dalam proses perbandingan kondisi $(\overline{Q}[i,j] \neq \overline{T}[i,j])$ akan lebih sering terjadi daripada kondisi $(\overline{Q}[i,j] = \overline{T}[i,j])$.

Dari empat pendekatan di atas maka persamaan metrik (3.3) akan menjadi :

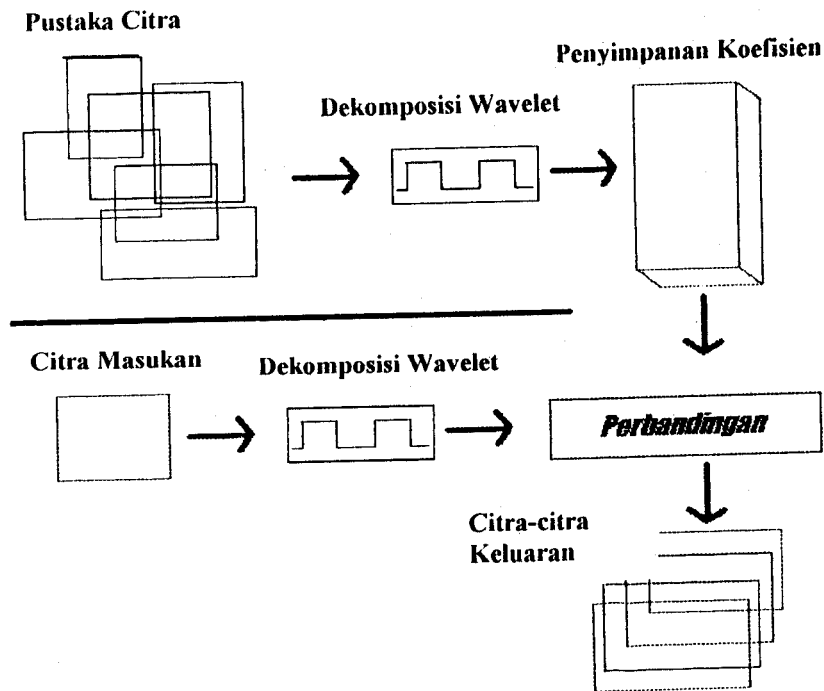
$$(3.4) \dots \dots \dots \|Q, T\| = w_0 |Q[0,0] - T[0,0]| - \sum_{i,j: Q[i,j] \neq 0} w_{bin(i,j)} (\overline{Q}[i,j] \neq \overline{T}[i,j])$$

yang merupakan suatu bentuk persamaan metrik yang dapat digunakan untuk mendapatkan nilai perbedaan antara citra masukan dan target dengan harga komputasi yang relatif murah.

3.3 Pencarian Citra Dengan Metode Dekomposisi Wavelet

Proses pencarian citra dengan metode dekomposisi wavelet merupakan metode pencarian citra yang memungkinkan pencarian dapat dilakukan dengan cepat. Hal ini sangat mungkin terjadi karena perbandingan antara citra masukan dan pustaka citra yang terdapat dalam basisdata pada proses pencarian tidak dilakukan secara langsung, perbandingan hanya dilakukan terhadap koefisien-koefisien wavelet terpilih hasil dekomposisi yang mewakili citra-citra yang terlibat dalam proses pencarian. Secara garis besar proses pencarian citra dengan metode dekomposisi wavelet dapat digambarkan pada gambar 3.1.

Proses diawali dengan melakukan dekomposisi wavelet terhadap seluruh citra yang ada pada basisdata citra guna mendapatkan koefisien-koefisien wavelet dari seluruh citra dan nilai rerata warnanya. Proses dekomposisi terhadap citra adalah dengan mengimplementasikan dekomposisi wavelet standard dua dimensi terhadap nilai-nilai struktur warna R,G,B untuk setiap piksel yang ada pada citra.



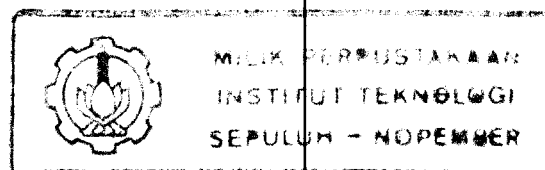
Gambar 3.1, Proses Pencarian Citra dengan Metode Dekomposisi Wavelet.

Proses tersebut dilakukan dengan mengimplementasikan dekomposisi wavelet satu dimensi terhadap seluruh nilai piksel untuk setiap baris citra dan selanjutnya diikuti dengan melakukan dekomposisi wavelet satu dimensi terhadap seluruh kolomnya. Proses ini merupakan implementasi dari persamaan wavelet dua dimensi (2.10).

Langkah selanjutnya adalah dengan melakukan dekomposisi yang sama terhadap citra masukan guna mendapatkan rerata warna dan koefisien wavelet yang dapat mewakili citra masukan dalam proses pencarian.

Proses pencarian dilakukan dengan mengimplementasikan persamaan metrik (3.4) untuk melakukan perbandingan nilai-nilai koefisien hasil dekomposisi yang mewakili citra-citra yang terlibat dalam proses pencarian. Hal ini dilakukan untuk

mendapatkan nilai-nilai metrik yang merupakan nilai beda antara citra masukan dengan pustaka citra. Pustaka citra yang dianggap paling sesuai sebagai citra keluaran hasil pencarian adalah pustaka citra yang memberikan nilai beda terkecil hasil perbandingan terhadap citra masukan.



BAB IV

PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab ini akan dijelaskan mengenai perancangan dan implementasi dari perangkat lunak yang dibuat meliputi gambaran perangkat lunak yang dibuat, struktur data, dan beberapa fungsi dan prosedur yang penting yang merupakan penerapan dasar teori yang telah diuraikan pada bab II dan bab III di atas. Program yang dibuat menggunakan bahasa pemrograman Borland C++ Builder dibawah sistem operasi Windows 95 ke atas.

4.2 Sistem Perangkat Lunak dan Tujuannya

Sasaran dari perangkat lunak ini adalah untuk mengimplementasikan proses pencarian citra terhadap basisdata citra untuk mendapatkan citra yang bersesuaian dengan cepat. Dalam pengoperasiannya perangkat lunak membutuhkan sebuah basisdata citra yang digunakan sebagai basisdata target pada proses pencarian. Citra yang digunakan adalah citra dengan tipe BMP sebagai citra masukan dan pustaka citra yang terdapat pada basisdata. Pada sebuah basisdata target perlu dilakukan proses dekomposisi terhadap seluruh citranya untuk mendapatkan rerata warna dan koefisien-koefisien wavelet untuk setiap citra. Nilai-nilai koefisien tersebut tidaklah seluruhnya perlu digunakan, namun dalam hal ini hanya dipilih berapa nilai-nilai yang dianggap paling dapat memawiliki keberadaan citranya dan nilai rerata pikselnya untuk semua struktur warna R/G/B. Pada akhirnya untuk mengoptimalkan program, nilai-nilai

koefisien wavelet tidak perlu disimpan, namun hanya nilai rerata, posisi piksel wavelet terpilih dan tanda nilainya (positif / negatif). Penyimpanan dilakukan dengan memanfaatkan sebuah file biner. Bagian proses di atas disebut sebagai bagian praproses dari proses pencarian. Untuk melakukan pencarian, pengguna memasukkan sebuah citra masukan bertipe citra BMP atau berupa sketsa yang dibuat langsung pada editor citra yang disediakan. Sebelum proses pencarian, dilakukan juga dilakukan proses dekomposisi terhadap citra masukan yang diberikan untuk mendapatkan nilai rerata warna dan koefisien-koefisien waveletnya dan selanjutnya akan digunakan dalam proses perbandingan yang dilakukan pada proses pencarian. Hasil perbandingan berupa dua puluh citra dari basisdata yang memberikan nilai beda terkecil hasil dari proses perbandingan dengan citra masukan. Citra-citra keluaran akan ditampilkan pada sebuah *form* secara terurut berdasarkan nilai bedanya.

4.3 Perancangan dan Implementasi Perangkat Lunak

Setiap perangkat lunak selalu terdiri dari dua bagian penting yang meliputi data dan proses yang mengolah data yang ada sesuai dengan fungsinya. Pada bagian ini pembahasan diawali dengan menjelaskan alur program yang dibuat dengan menggunakan *data flow diagram*. Program yang dibuat menggunakan bahasa pemrograman visual. Pada pemrograman visual aplikasi disusun dari banyak *form* dan setiap *form* merupakan sebuah kelas dengan obyek-obyek dan prosedur-prosedur yang dimilikinya, maka dalam penjelasan ini penulis

menjelaskan struktur data dan implementasi prosedur proses dari aplikasi yang dibuat berdasarkan *form-form* yang ada dalam aplikasi.

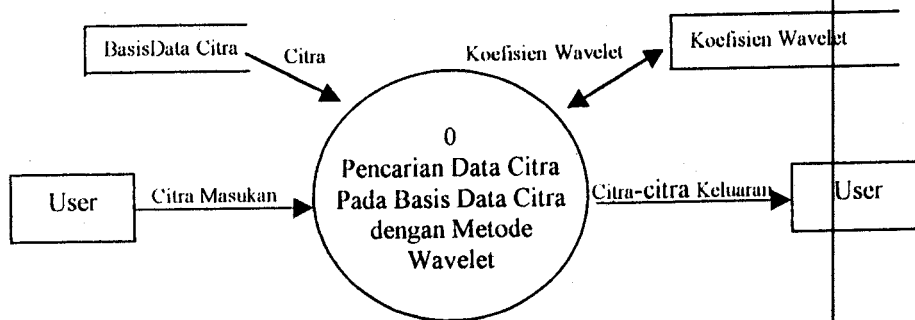
Pada pembuatan perangkat lunak ini ada dua bagian program yang dibuat yang meliputi bagian praproses pencarian dan bagian proses pencarian.

Bagian praproses berfungsi untuk mempersiapkan file-file koefisien wavelet yang merupakan hasil transformasi wavelet terhadap citra-citra yang ada pada file-file basisdata citra dengan menggunakan fungsi dekomposisi *haar wavelet* dua dimensi. Selanjutnya disimpan pada sebuah file nilai rerata warna dan koefisien-koefisien wavelet yang dianggap paling mewakili keberadaan citra untuk setiap struktur warna R,G,B. File koefisien ini digunakan untuk mewakili file basisdata citranya dalam proses pencarian.

Pada bagian kedua yakni bagian proses pencarian, sebelum dilakukan pencarian diawali dengan melakukan dekomposisi wavelet terhadap citra masukan untuk mendapatkan rerata warna dan koefisien-koefisien wavelet yang dianggap dapat mewakili keberadaan citra masukan. Selanjutnya proses dilanjutkan dengan memperbandingkan nilai-nilai itu dengan nilai-nilai yang terdapat pada file koefisien dengan persamaan perbandingan metrik (3.4) untuk mendapatkan citra-citra yang dianggap memiliki perbedaan paling kecil sebagai citra keluaran.

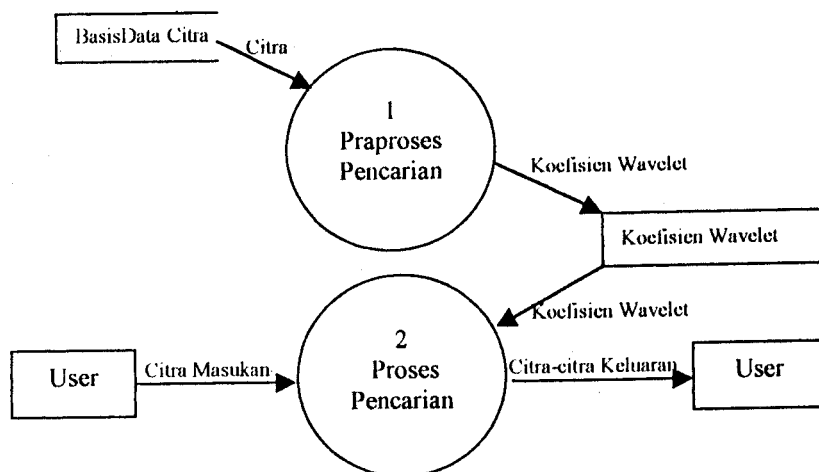
4.3.1 Data Flow Diagram

Data Flow Diagram (DFD) digunakan untuk menjelaskan alur data pada aplikasi. *Context diagram* dari aplikasi yang dibuat ditampilkan pada gambar 4.1.



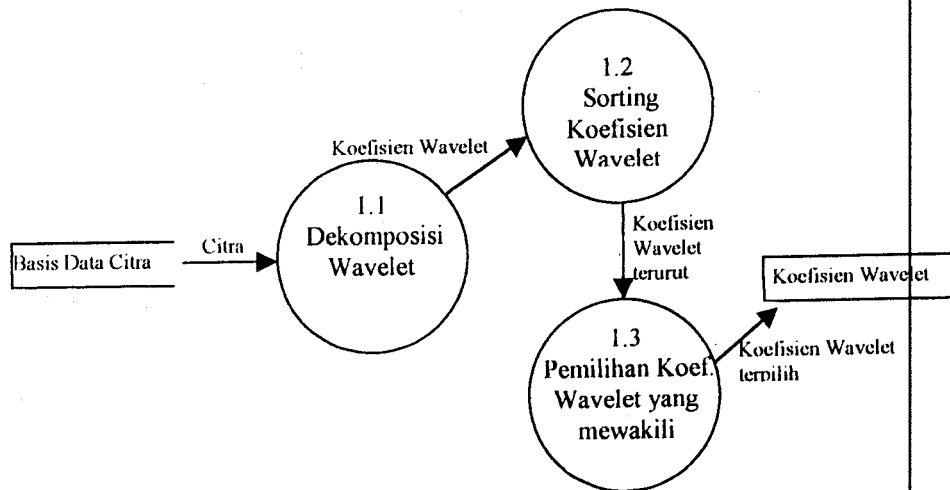
Gambar 4.1 Data Flow Diagram tingkat 0 Aplikasi Pencarian Data Citra Pada Basis Data Citra Dengan Metode Wavelet.

Data Flow Diagram tingkat 0 di atas merupakan penggambaran secara umum dari alur data pada aplikasi yang dibuat. Selanjutnya untuk lebih memperjelas DFD di atas akan dijabarkan pada tingkat-tingkat yang lebih terperinci sebagai berikut .

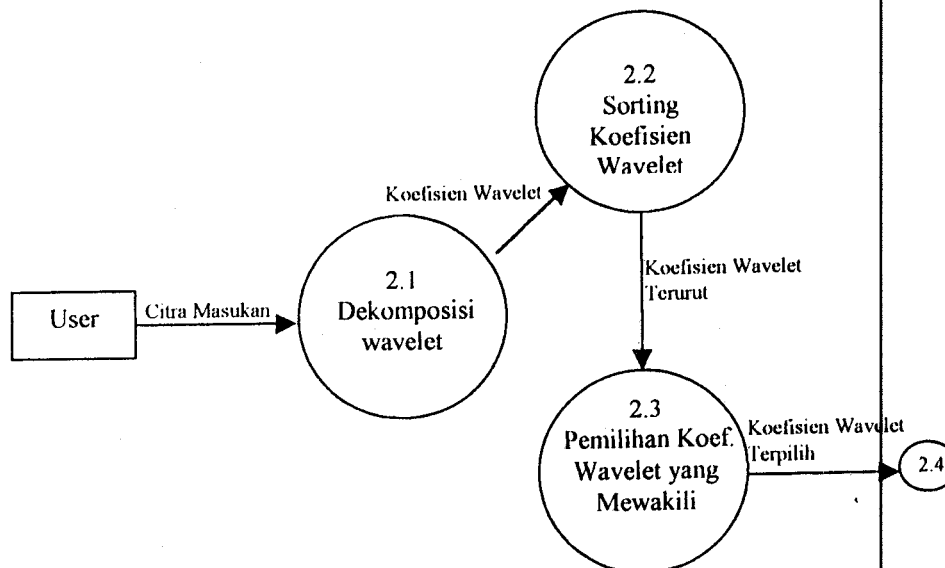


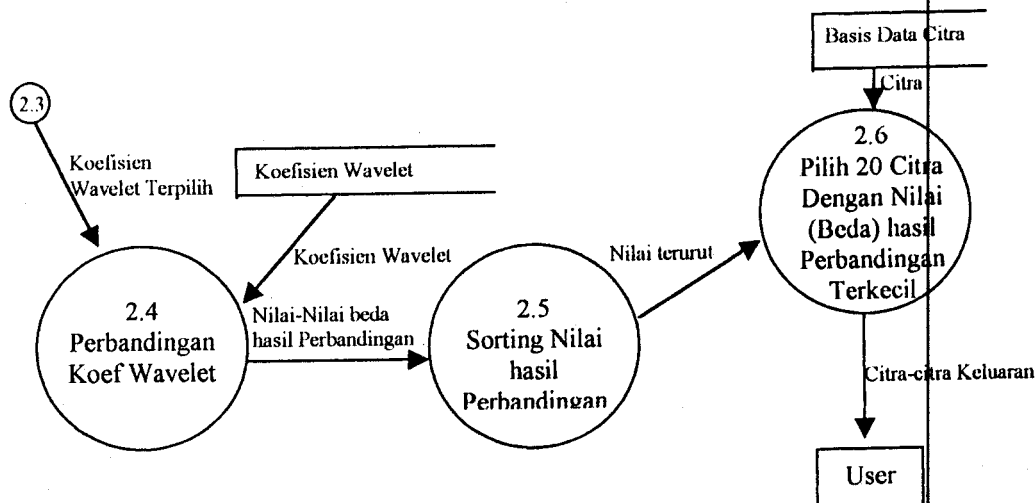
Gambar 4.2 Data Flow Diagram Tingkat 1, proses-proses pada aplikasi pencarian data citra

Selanjutnya setiap bagian proses pada DFD di atas dapat terus diuraikan menjadi lebih terperinci pada DFD tingkat 2 dibawah ini.



Gambar 4.3 Data Flow Diagram tingkat 2 , Praproses Pencarian





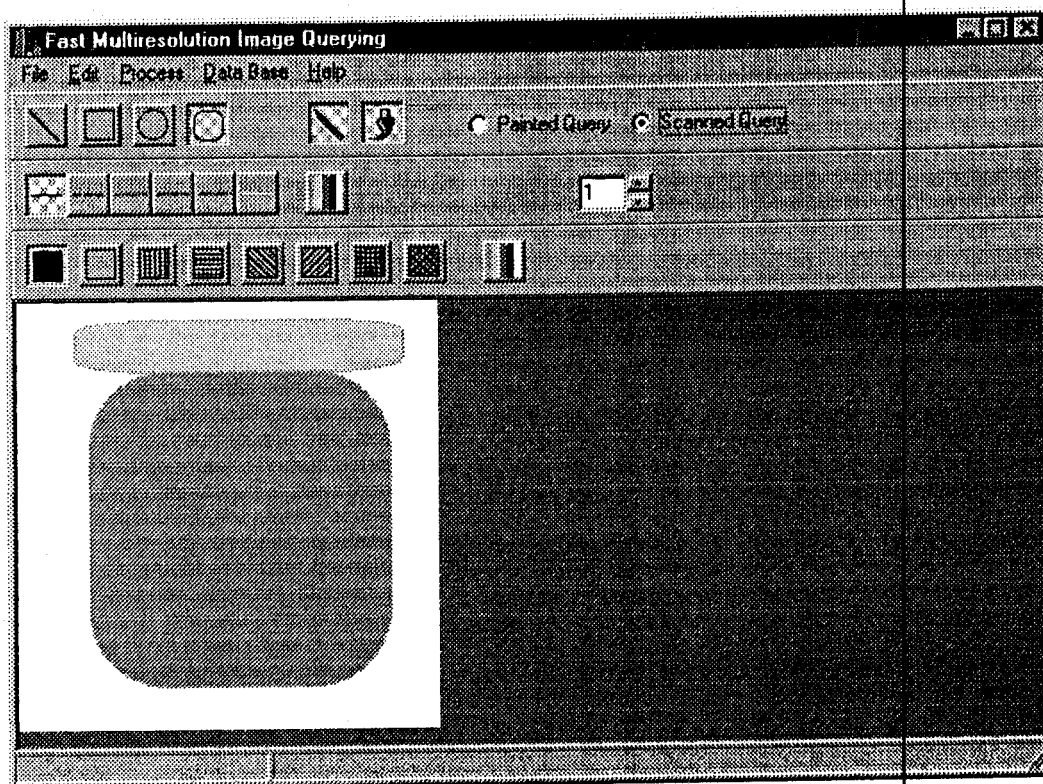
Gambar 4.4 . Data Flow Diagram tingkat 2, Proses Pecarian Citra Pada Basis Data Citra

4.3.2 Perancangan Form dan Datanya

Aplikasi yang dibuat terdiri atas banyak *form*, dimana setiap *form* memiliki fungsi-fungsi tertentu sesuai dengan struktur data, fungsi-fungsi, dan prosedur-prosedur yang dimilikinya. Pada aplikasi ini *form-form* yang dimiliki adalah sebagai berikut :

4.3.2.1 Form Utama

Form ini merupakan *form* utama dari aplikasi yang dibuat. Dimana seluruh proses yang ada pada aplikasi terorganisir pada *form* ini . Tampilan muka dari *form* ini adalah sebagai berikut :



Gambar 4.5 *Form Utama*

Form ini sebagai tampilan utama dari aplikasi yang dibuat dimana semua fungsi yang ada pada aplikasi dapat digunakan melalui *form* ini. Diantaranya fungsi-fungsi itu adalah :

- Melakukan proses pencarian data citra pada sebuah basis data citra.
- Membuat sebuah sketsa citra yang berfungsi sebagai citra masukan pada pencarian citra.
- Melakukan praproses terhadap sebuah basis data citra.
- Persiapan basisdata target.
- Pengolahan basisdata citra.

Struktur data / kelas utama yang dimiliki oleh *form* ini adalah sebagai berikut :

```
class TMainFrm : public TForm
```

```

{
__published:    // IDE-managed Components
    void __fastcall Exit1Click(TObject *Sender);
    void __fastcall FormResize(TObject *Sender);
    void __fastcall FormCreate(TObject *Sender);
    void __fastcall Open1Click(TObject *Sender);
    void __fastcall Add1Click(TObject *Sender);
    void __fastcall Direktori1Click(TObject *Sender);
    void __fastcall Preproses1Click(TObject *Sender);
    void __fastcall Search1Click(TObject *Sender);
    void __fastcall OpenKoef1Click(TObject *Sender);
    void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
    void __fastcall Delete1Click(TObject *Sender);
    void __fastcall About1Click(TObject *Sender);
    void __fastcall RoundRectButtonClick(TObject *Sender);
    void __fastcall BrushButtonClick(TObject *Sender);
    void __fastcall PenButtonClick(TObject *Sender);
    void __fastcall EllipseButtonClick(TObject *Sender);
    void __fastcall RectangleButtonClick(TObject *Sender);
    void __fastcall LineButtonClick(TObject *Sender);
    void __fastcall SetPenClick(TObject *Sender);
    void __fastcall BrushColorClick(TObject *Sender);
    void __fastcall New1Click(TObject *Sender);
    void __fastcall Save1Click(TObject *Sender);
    void __fastcall SaveAs1Click(TObject *Sender);
    void __fastcall Print1Click(TObject *Sender);
    void __fastcall SetBrushStyle(TObject *Sender);
    void __fastcall ImageMouseDown(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y);
    void __fastcall ImageMouseMove(TObject *Sender, TShiftState Shift, int X,
    int Y);
    void __fastcall ImageMouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y);
    void __fastcall PaintedClick(TObject *Sender);
    void __fastcall ScannedClick(TObject *Sender);
    void __fastcall PenSizeChange(TObject *Sender);
    void __fastcall PenColorClick(TObject *Sender);
    void __fastcall Copy1Click(TObject *Sender);
    void __fastcall Cut1Click(TObject *Sender);
    void __fastcall Paste1Click(TObject *Sender);

```

private:

```

AnsiString CurrentFile;
Myhand hand[180];
int NumCof;
DArray *DaArr1,*DaArr2,*DaArr3,*HandS,*Score;
float T[3],AR,AG,AB,WR[6],WG[6],WB[6];
float **DataG,**DataR,**DataB;
Mylist **MlistP,**MlistM;
float *AveR,*AveG,*AveB;

```

public:

```

    __fastcall TMainFrm(TComponent* Owner);
    void __fastcall DecomposeImage(Graphics::TBitmap *);
    void __fastcall DecomposeArray(float *&,int);

```

```

void __fastcall HeapSort(DArray *DaArr,unsigned long );
void __fastcall SaveKoeff();
void __fastcall OpenKoeff();
void __fastcall prepare();
void __fastcall GetScore();
void __fastcall SetResult();
void __fastcall SetDB();
int __fastcall bin(int , int );
unsigned short __fastcall SetMatrix(unsigned short index);
void __fastcall FreeMatrix();
void __fastcall DrawShape(TPoint TopLeft, TPoint BottomRight, TpenMode
AMode);
bool Drawing;
TPoint Origin, MovePt;
TDrawingTool DrawingTool;
long NumOfRec;
bool SettingUp,HaveKof,HaveDek;
Graphics::TBitmap *MyImage;
};

```

Kelas di atas memiliki prosedur-prosedur penting yang fungsinya akan dijelaskan dibawah ini :

1. Prosedur *Dekompose Image*

Procedur ini dituliskan sebagai :

*void __fastcall DecomposeImage(Graphics::TBitmap *)*; berfungsi untuk melakukan proses dekomposisi wavelet dua dimensi terhadap sebuah citra guna mendapatkan rerata warna dan koefisien-koefisien wavelet untuk setiap struktur warna R,G B yang dapat mewakili citra pada proses pencarian citra. Prosedur ini menerima masukan data bitmap yang selanjutnya nilai-nilai piksel citra tersebut didekomposisi. Prosedur ini disusun dengan mengimplementasikan *pseudocode* dibawah ini.

```

Proc DekomposeImage(T : array[0...r-1,0..r-1] of color )
  For row ← 1 to r do :
    DecomposeArray(T[row,0..r-1])
  EndFor

```

```

For col  $\leftarrow$  1 to r do :
    DecomposeArray( $T[0..r-1, col]$ )
EndFor
EndProc

```

Pada *pseudocode* diatas terlihat bahwa prosedur ini memanfaatkan satu prosedur lain, yakni prosedur *DecomposeArray($T[0..r-1, col]$)* yang melakukan fungsi dekomposisi wavelet satu dimensi terhadap nilai-nilai warna setiap baris dan kolom dari citra masukan pada prosedur *DekomposeImage($T : array[0...r-1, 0..r-1]$ of color.*

2. Prosedur Dekompose Array

Prosedur ini dituliskan sebagai :

*void __fastcall DecomposeArray(float *&, int)* ;. Seperti telah dijelaskan di atas prosedur ini berfungsi untuk melakukan Dekomposisi Wavelet satu dimensi terhadap setiap baris dan kolom nilai-nilai piksel citra masukan pada prosedur *DecomposeImage(Graphics::TBitmap *)*. Prosedur ini menerima nilai array dari warna selanjutnya melakukan dekomposisi wavelet satu dimensi terhadap nilai array tersebut. Prosedur ini disusun dengan mengimplementasikan *pseudocode* dibawah ini.

```

Proc DekomposisiArray( $A : array [0..h-1]$  of color)
     $A \leftarrow A / \sqrt{h}$ 
    while  $h > 1$  do
         $h \leftarrow h/2$ 
        for  $i \leftarrow 0$  to  $h-1$  do
             $A'[i] \leftarrow (A[2i] + A[2i+1]) / \sqrt{2}$ 
             $A'[h+i] \leftarrow (A[2i] - A[2i+1]) / \sqrt{2}$ 
        endfor
         $A \leftarrow A'$ 
    endwhile
end.

```

3. Prosedur *Heap Sort*

Prosedur ini dituliskan sebagai :

*voidfastcall HeapSort(DArray *DaArr,unsigned long)*, berfungsi melakukan pengurutan terhadap nilai-nilai data masukan. Nilai-nilai data yang digunakan berstruktur data khusus sesuai dengan kebutuhan aplikasi. Struktur data tersebut berbentuk *Darray* dengan struktur sebagai berikut :

```
enum minplus {minus,plus};
struct DArray
{
    minplus type;
    unsigned i,j;
    float item;
};
```

Pengurutan dilakukan berdasarkan nilai dari variabel *item* yang bertipe data *float*. Tipe data *Darray* di atas digunakan sebagai tipe data untuk menyimpan nilai koefisien wavelet hasil dekomposisi wavelet dua dimensi terhadap nilai-nilai warna untuk setiap piksel yang telah dimutlakkan. Variabel *i* dan *j* digunakan untuk menyimpan posisi piksel pada citra dan variabel *type* digunakan untuk menyimpan tipe angka positif atau negatif sebelum nilai dimutlakkan. Hal-hal tersebut di atas diperlukan untuk keperluan proses pencarian.

Implementasi dari algoritma *heap sort* pada aplikasi yang dibuat adalah sebagai berikut :

```
Proc HeapSort(T : Array[1 .. n] of item)
{
    for (i = 2; i <= n; i++)
    {
        elt = T[i];
```



```

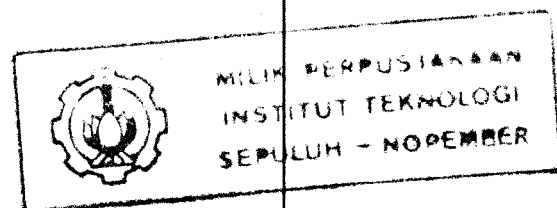
s = i;
f = s/2;
while ( (s>1) && (T[f]>elt) )
{
    T[s] = T[f];
    s = f;
    if (s>1) f = s/2;
    else f = 1;
}
T[s] = elt;
}
for (i=n-1; i>1; i--)
{
    ivalue = T[i];
    T[i] = T[1];
    f = 1;
    if (i==2) s = 0;
    else s=2;
    if ( (i>3) && (T[3]<T[2]) ) s = 3;
    value = T[s];
    while ( (s>0) && (ivalue>value) )
    {
        T[f] = T[s];
        f = s;
        s = 2*f;
        if ( (s+1) <= (i-1) )
        if ( T[s] > T[s+1] ) ++s;
        if ( s<i ) value = T[s];
        else
        {
            s = 0;
            value = T[0];
        }
    }
    T[f] = ivalue;
}
}

```

4. Prosedur *Get Score*

Prosedur ini dituliskan sebagai :

void __fastcall GetScore(); Berfungsi untuk melakukan perbandingan antara koefisien wavelet dan rerata warna dari citra masukan dan pustaka citra untuk



mendapatkan nilai bedanya. Prosedur ini merupakan implementasi dari persamaan metrik (3.4). Nilai-nilai beda ini digunakan untuk mendapatkan pustaka citra yang dianggap paling mendekati sebagai citra-citra keluaran, yakni yang menghasilkan nilai beda terkecil. Prosedur ini diimplementasikan dengan algoritma sebagai berikut :

Prosedur GetScore()

Var T : Pustaka citra

Q : Citra Masukan

For i ← 0 to Banyak Citra dalam basisdata-1 do:

Inisialisasi Score[i] = 0 untuk semua i

Endfor

For Semua Struktur warna c do:

For semua basisdata citra T do:

Score[indek[T]] += $w^c[0] * |Q^c[0,0] - T^c[0,0]|$

Endfor

q ← nilai koefisien wavelet terpilih citra masukan

For Semua Nilai q[i,j] <> 0 do:

If $q^c[i,j] > 0$ then

list ← $D^c_+[i,j]$

else

list ← $D^c_-[i,j]$

Endif

For Semua nilai l pada list do:

Score[l] -= $w^c[\text{bin}(i,j)]$

Endfor

Endfor

Endfor

End.

Pada *pseudocode* di atas digunakan beberapa variabel yang fungsi masing-masingnya akan dijelaskan di bawah ini :

- ***Score[]***: Digunakan untuk menyimpan nilai beda antara citra masukan dan pustaka citra. ***Score[i]*** untuk target ke *i* pada basisdata.

- $D^c_+ [i,j]$: Merupakan sebuah matrik dua dimensi. $D^c_+ [i,j]$ berisi daftar indek dari pustaka citra yang memiliki nilai koefisien wavelet positif terpilih pada posisi baris i dan kolom j .
- $D^c_- [i,j]$: Merupakan sebuah matrik dua dimensi. $D^c_- [i,j]$ berisi daftar indek dari pustaka citra yang memiliki nilai koefisien wavelet negatif terpilih pada posisi baris i dan kolom j .
- *list* : Berisikan daftar indek dari pustaka citra yang memiliki nilai koefisien wavelet negatif/positif terpilih pada posisi baris i dan kolom j .

Pada implementasi program variabel $D^c_+ [i,j]$ dan $D^c_- [i,j]$ digunakan untuk optimasi proses pencarian dan diimplementasikan dalam satu bentuk struktur data sebagai berikut:

*Mylist **MlistP, **MlistM;*

Dimana bentuk data *Mylist* adalah :

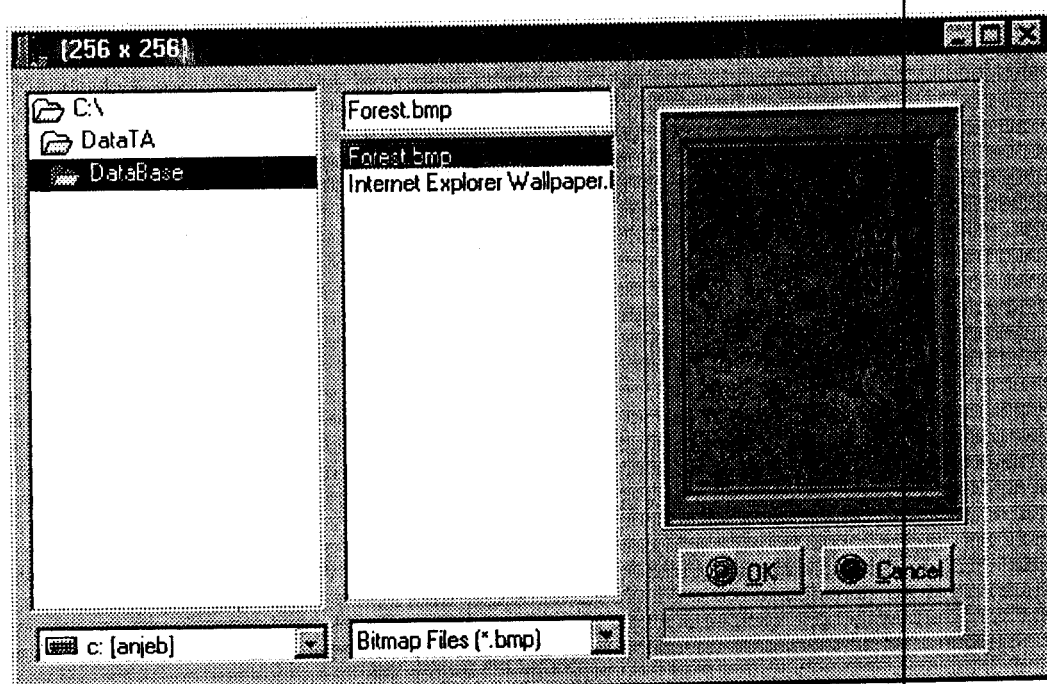
```
struct Tdata
{
    unsigned short Item;
    Tdata *Next;
};
class Mylist
{
private:
public:
    void __fastcall AddR(unsigned short );
    void __fastcall AddG(unsigned short );
    void __fastcall AddB(unsigned short );
    void __fastcall FreeData();
    Mylist();
    Tdata *DataR, *HeadR, *DataG, *HeadG, *DataB, *HeadB;
};
```

MlistP adalah implementasi dari D^c_+ [,] dan *MlistM* adalah implementasi dari D^c_- [,]

Prosedur-prosedur dan struktur data di atas adalah prosedur-prosedur dan struktur data penting yang ada pada *form* utama dan berhubungan langsung dengan proses pencarian citra pada basisdata citra dengan metode dekomposisi wavelet. Sedangkan prosedur-prosedur dan properti-properti lainnya yang terdapat pada *form* tersebut merupakan prosedur-prosedur dan properti-properti tambahan yang keberadaannya dikarenakan penggunaan fasilitas-fasilitas *form* yang diberikan dalam pemrograman visual yang berfungsi sebagai penunjang dari fungsi kerja *form* secara keseluruhan.

4.3.2.2 Form ImageViewer

Form ini berfungsi untuk membuka sebuah citra bitmap dari direktorinya untuk digunakan sebagai citra masukan pada aplikasi yang dibuat. *Form* ini dilengkapi oleh sebuah fasilitas viewer citra untuk mengetahui apakah citra yang akan dibuka sesuai dengan yang diinginkan oleh pengguna. Tampilan muka dari *form* ini adalah :



Gambar 4.6 *Form Image Viewer*

Struktur data / kelas yang dimiliki oleh *form* ini adalah sebagai berikut :

class TImageForm : public TForm

```
{
    __published:
        TBevel *Bevel1;
        TDirectoryListBox *DirectoryListBox1;
        TDriveComboBox *DriveComboBox1;
        TEdit *FileEdit;
        TPanel *Panel1;
        TImage *Image1;
        TBitBtn *ViewBtn;
        TFilterComboBox *FilterComboBox1;
        TFileListBox *FileListBox1;
        TBitBtn *BitBtn1;
        TProgressBar *Progress;
        void __fastcall FormCreate(TObject *Sender);
        void __fastcall ViewBtnClick(TObject *Sender);
        void __fastcall FileEditKeyPress(TObject *Sender, Char &Key);
        void __fastcall FileListBox1Click(TObject *Sender);
        void __fastcall CancelBtnClick(TObject *Sender);
        void __fastcall FormClose(TObject *Sender, TCloseAction &Action);

    public:
        AnsiString FormCaption;
        AnsiString FileExt;
        virtual __fastcall TImageForm(TComponent *Owner);
};
```

Sesuai dengan fungsinya, *form* ini digunakan oleh pengguna untuk membuka sebuah file citra untuk digunakan sebagai file masukan pada proses pencarian citra, karena itu pada *form* ini tidak terdapat prosedur penting yang berhubungan langsung dengan proses pencarian dengan metode dekomposisi wavelet. Prosedur utama dalam *form* ini adalah prosedur *void __fastcall ViewBtnClick(TObject *Sender)*. Prosedur ini akan otomatis dipanggil begitu pengguna aplikasi melakukan penekanan pada tombol *OK* yang ada pada *form*. Hal-hal yang dapat dilakukan melalui *form* ini adalah :

- Melakukan penelusuran terhadap letak sebuah file citra bitmap.
- Membuka file bitmap yang dikehendaki sebagai citra masukan pada proses pencarian citra pada database citra.

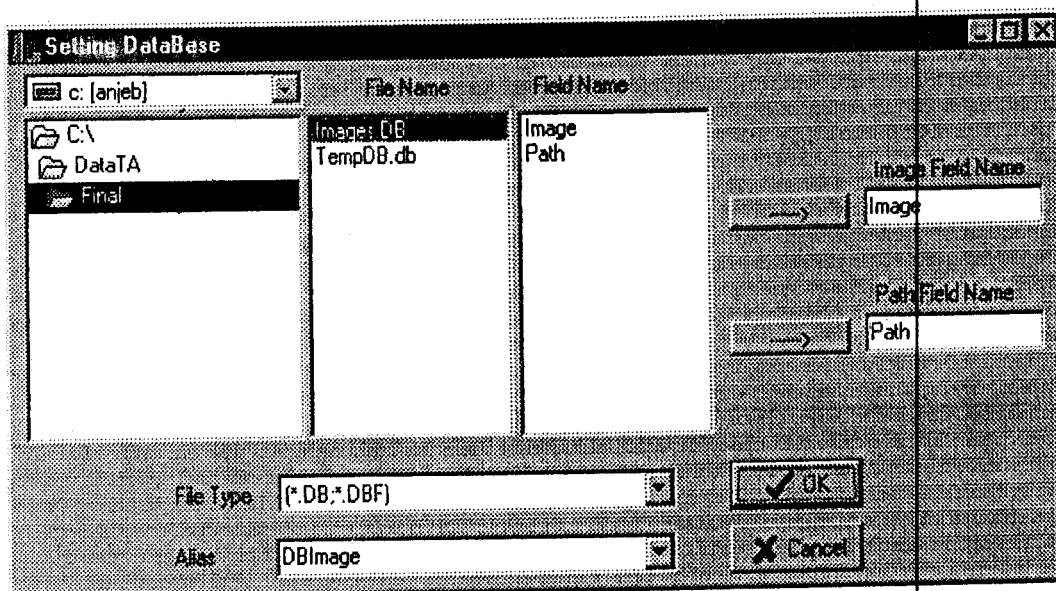
Apabila pengguna telah memutuskan untuk membuka sebuah file citra dengan menekan tombol *OK* , maka prosedur *void __fastcall ViewBtnClick(TObject *Sender)* akan melakukan hal-hal di bawah ini :

- Mengirimkan citra terhadap prosedur dekomposisi wavelet yang terdapat pada *form* utama untuk dilakukan proses dekomposisi wavelet.
- Mengirimkan citra untuk ditampilkan pada editor citra yang terdapat pada *form* utama.

Prosedur-prosedur dan properti-properti yang ada pada *class* dari *form* ini merupakan prosedur dan properti yang ditimbulkan oleh penggunaan fasilitas-fasilitas dari pemrograman visual yang dipilih karena kebutuhan untuk menunjang fungsi kerja dari *form* yang dibuat.

4.3.2.3 Form Set Basisdata

Form ini berfungsi bagi pengguna untuk melakukan *setting* terhadap basis data yang akan digunakan sebagai basisdata target pada proses pencarian citra. Dengan demikian pada aplikasi ini dimungkinkan untuk melakukan pencarian terhadap basisdata citra apapun juga. Tampilan muka dari form ini adalah :



Gambar 4.7 Form Setting Basisdata

Struktur data / kelas yang dimiliki oleh form ini adalah sebagai berikut :

```
class TSetDataFrm : public TForm
{
    __published: // IDE-managed Components
        TListBox *ListBox1;
        TFilterComboBox *FilterComboBox1;
        TEdit *Edit1;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TDirectoryListBox *DirectoryListBox1;
        TBitBtn *OkBtn;
        TBitBtn *CancelBtn;
        TLabel *Label4;
        TLabel *Label5;
        TFileListBox *FileListBox1;
        TLabel *Label6;
        TTable *Table1;
        TComboBox *ComboBox1;
        TEdit *Edit2;
```

```

TBitBtn *SetImageBtn;
TBitBtn *SetPathBtn;
TLabel *Label7;
TDriveComboBox *DriveComboBox1;
void __fastcall FormCreate(TObject *Sender);
void __fastcall CancelBtnClick(TObject *Sender);
void __fastcall FileListBox1Click(TObject *Sender);
void __fastcall ComboBox1Click(TObject *Sender);
void __fastcall DirectoryListBox1Click(TObject *Sender);
void __fastcall OkBtnClick(TObject *Sender);
void __fastcall SetImageBtnClick(TObject *Sender);
void __fastcall SetPathBtnClick(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
private: // User declarations
    bool SetAlias;
public: // User declarations
    __fastcall TSetDataFrm(TComponent* Owner);
};

```

Sebelum sebuah file basisdata citra digunakan sebagai file target pencarian, file tersebut harus diset terlebih dahulu sebagai basisdata target dari aplikasi. Proses ini dilakukan untuk mengetahui properti dari basisdata yang digunakan untuk keperluan proses. Properti-properti yang dibutuhkan antara lain:

- Letak direktori/alias dari basisdata.
- Nama file basisdata.
- Nama field citra yang digunakan.
- Nama field path dari letak citra pada basis data untuk keperluan informasi.

Pada aplikasi ini bentuk struktur standard file basisdata yang digunakan adalah sebagai berikut :

- File basisdata harus memiliki minimal satu buah field citra bertipe data standard Borland *TgraphicField*.
- File basisdata harus memiliki minimal satu buah field path letak dari citra bertipe data standard Borland *TStringField*. Field ini digunakan sebagai

informasi bagi pengguna agar mengetahui letak asal dari citra yang ada pada basisdata.

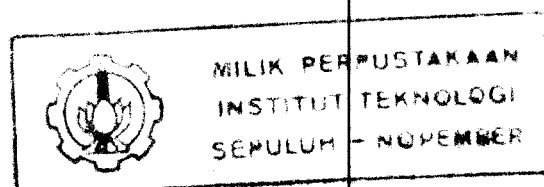
- Basis data boleh memiliki jenis field lainnya dan tidak akan mempengaruhi kinerja dari sistem.

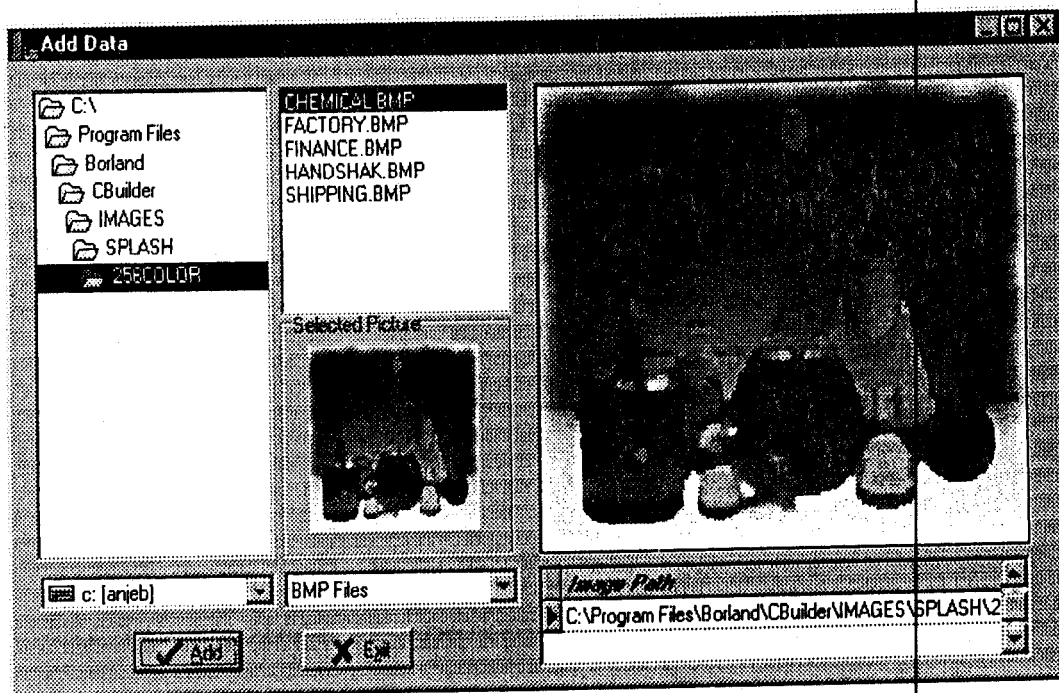
Apabila seluruh properti dari sebuah basis data telah diberikan melalui *form* ini dan telah memenuhi kriteria, maka seluruh proses yang dilakukan pada aplikasi ini diarahkan terhadap basisdata tersebut sebagai targetnya.

Seperti pada *form* sebelumnya, prosedur-prosedur dan properti -properti yang ada pada kelas dari *form* ini merupakan prosedur dan properti yang ditimbulkan oleh penggunaan fasilitas-fasilitas dari pemrograman visual yang dipilih karena kebutuhan untuk menunjang fungsi kerja dari *form* yang dibuat.

4.3.2.4 Form Tambah Data Citra Basisdata

Aplikasi ini menyediakan fasilitas bagi pengguna untuk melakukan penambahan data terhadap basisdata yang telah diset sebagai basisdata target pencarian. Proses tersebut dilakukan *form* ini. Selain dilakukan penambahan terhadap basisdata, apabila penambahan data dilakukan melalui *form* ini secara otomatis aplikasi akan melakukan penambahan data koefisien wavelet terhadap file koefisien wavelet dari basisdata tersebut dengan terlebih dahulu melakukan dekomposisi wavelet terhadap file citra yang ditambahkan terhadap basisdata. Tampilan muka dari *form* ini adalah :





Gambar 4.8 Form Tambah Data

Struktur data / kelas yang dimiliki oleh *form* ini adalah sebagai berikut :

```
class TAddData : public TForm
{
    __published: // IDE-managed Components
        TDBImage *DBImage1;
        TTable *Table1;
        TDBGrid *DBGrid1;
        TFileListBox *FileListBox;
        TDriveComboBox *DriveBox;
        TDirectoryListBox *DirListBox;
        TFilterComboBox *FilterComboBox1;
        TGroupBox *GroupBox1;
        TImage *Image1;
        TBitBtn *AddButton;
        TBitBtn *BitBtn2;
        TDataSource *DataSource1;
        TTable *Table2;
        TGraphicField *Table2Image;
        void __fastcall AddButtonClick(TObject *Sender);
        void __fastcall FormShow(TObject *Sender);
        void __fastcall BitBtn2Click(TObject *Sender);
        void __fastcall FileListBoxClick(TObject *Sender);
        void __fastcall Image1DbClick(TObject *Sender);
        void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
        void __fastcall DBImage1Click(TObject *Sender);
    private: // User declarations
    public: // User declarations
        __fastcall TAddData(TComponent* Owner);
};
```

Melalui *form* ini pengguna aplikasi dapat melakukan penelusuran terhadap letak file citra yang diinginkan dan melalui *form* ini juga pengguna dapat melihat seluruh data citra yang ada pada basisdata. Sebelum melakukan penambahan sebuah citra ke dalam basisdata, citra yang dipilih ditampilkan terlebih dahulu pada sebuah *viewer* citra kecil guna memastikan apakah citra yang dipilih sesuai dengan yang diinginkan. Penekanan tombol *Add* akan menambahkan data citra terpilih ke basis data dan penambahan data koefisien wavelet terhadap file koefisien dari basisdata target. Prosedur-prosedur dan properti-properti yang ada pada kelas dari *form* ini juga merupakan prosedur dan properti yang ditimbulkan oleh penggunaan fasilitas-fasilitas dari pemrograman visual yang dipilih karena kebutuhan untuk menunjang fungsi kerja dari *form* yang dibuat.

Form-form di atas adalah *form-form* penting dari aplikasi yang dibuat yang berhubungan langsung dengan proses pencarian citra pada basisdata citra dengan menggunakan metode dekomposisi wavelet. Disamping *form-form* di atas, terdapat *form-form* lainnya yang merupakan pelengkap dari aplikasi dan tidak berhubungan langsung dengan proses pencarian. *Form-form* tersebut diantaranya adalah :

1. *Form* Tampilan Citra Hasil Pencarian.

Form ini berfungsi untuk menampilkan citra-citra keluaran dari proses pencarian. Citra keluaran yang diberikan adalah 20 citra pada basisdata yang memiliki nilai beda terkecil terhadap citra masukan.

2. *Form About*

Form ini berfungsi untuk menampilkan identitas dari pembuat aplikasi.

3. *Form New Image*

Form ini digunakan untuk masukan ukuran dari citra yang akan dibuat oleh pengguna sebagai citra masukan dalam bentuk sketsa.

BAB V

UJI COBA DAN PEMBAHASANNYA

Bab ini membahas hasil percobaan dan analisis dari perangkat lunak yang dibuat, yakni dengan melakukan pencarian data citra terhadap sebuah basisdata citra dengan metode dekomposisi wavelet. Pembahasan yang dilakukan meliputi keberhasilan pencarian citra terhadap basisdata target dan pengaruh-pengaruh dari perubahan citra masukan terhadap keberhasilan pencarian, serta pembahasan mengenai kecepatan pencarian dan faktor-faktor yang mempengaruhinya.

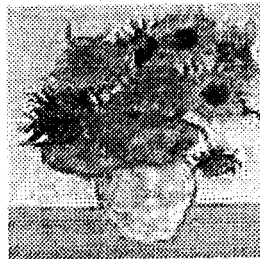
5.1 Uji Coba dan Analisisnya

Percobaan dilakukan dalam dua cara dengan tujuan yang berbeda. Cara pertama adalah percobaan dengan menggunakan sebuah citra masukan dan melakukan perubahan terhadap citra masukan yang diberikan. Percobaan tersebut dilakukan untuk mengetahui pengaruh perubahan tersebut terhadap keberhasilan pencarian. Percobaan kedua adalah melakukan percobaan pencarian secara berulang-ulang terhadap basisdata dengan citra masukan yang berbeda-beda untuk mengetahui tingkat keberhasilan pencarian dari perangkat lunak yang dibuat.

5.1.1 Uji Coba dengan Perubahan-Perubahan Pada Citra Masukan

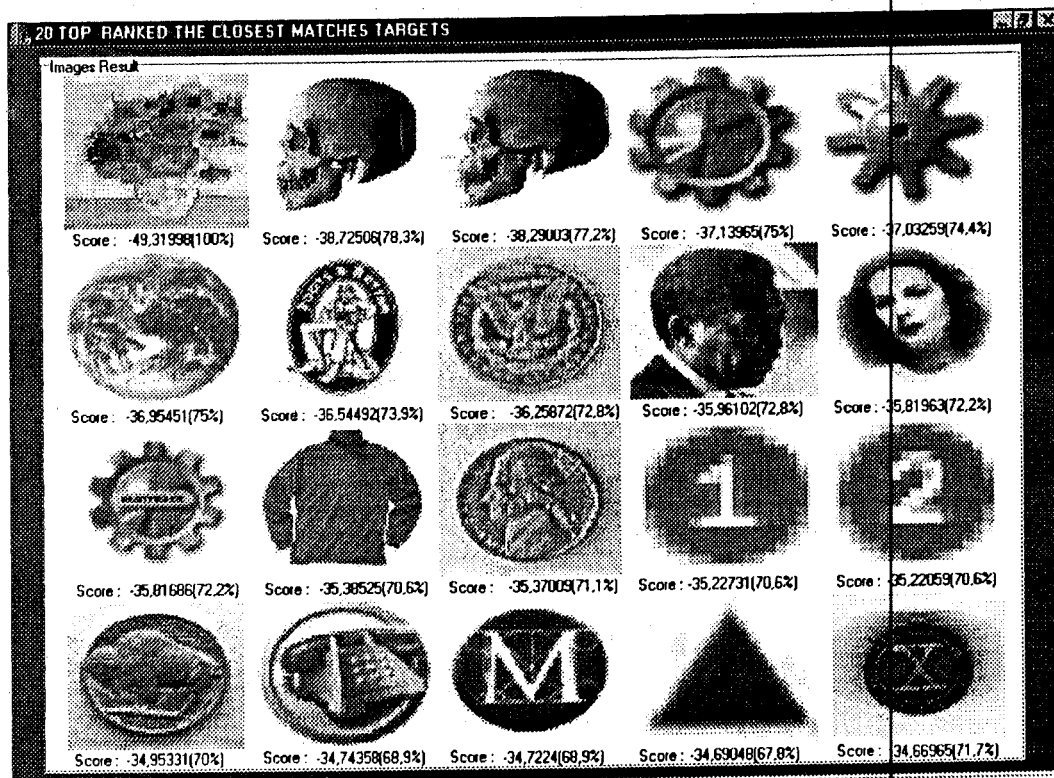
Uji coba ini dilakukan untuk mengetahui akibat dari perubahan-perubahan yang dilakukan terhadap citra masukan dan pengaruhnya terhadap keberhasilan pencarian. Perubahan-perubahan yang dilakukan meliputi perubahan tingkat

warna, perubahan warna citra ke dalam skala abu-abu (*grey scale*), pengaburan warna, perubahan gambar citra dengan menambah atau mengurangi informasi citra, dan penggunaan sketsa sebagai citra masukan. Percobaan diawali dengan menggunakan citra masukan berkeadaan normal seperti yang terdapat pada basisdata.



Gambar 5.1. Citra masukan pada percobaan pertama

Setelah dilakukan pencarian terhadap basisdata didapatkan citra-citra keluaran sebagai berikut :



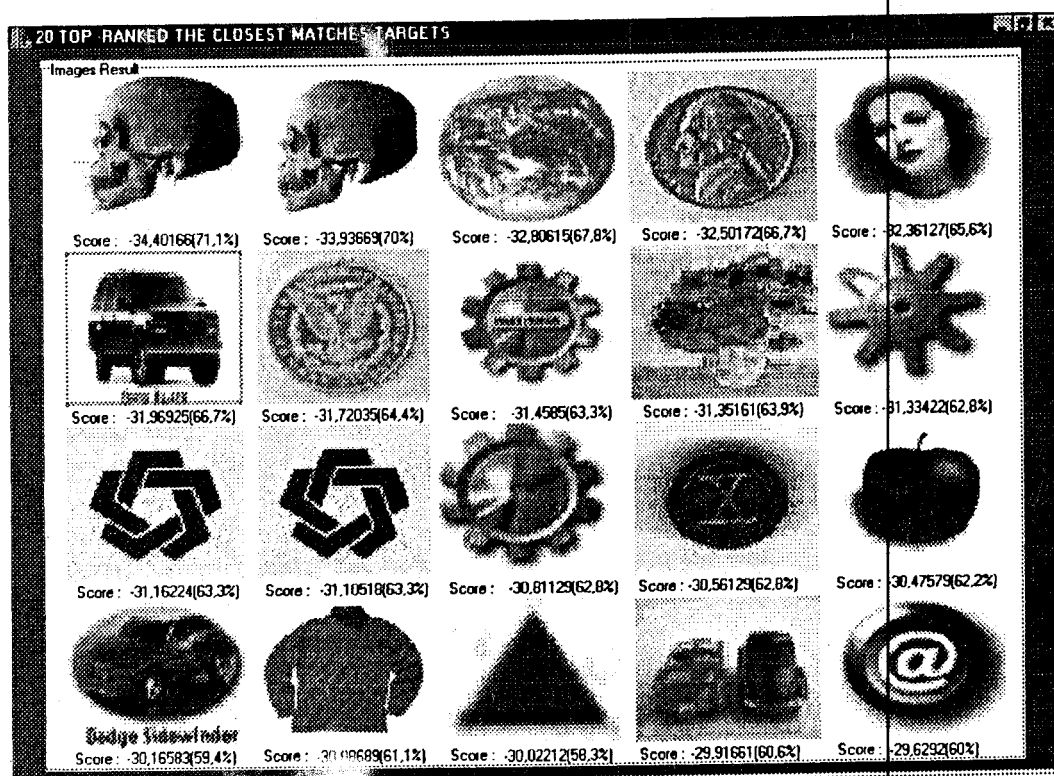
Gambar 5.2. Citra-citra keluaran pada percobaan pertama.

Percobaan kedua dilakukan dengan melakukan perubahan tingkat warna citra masukan. Citra masukan yang semula memiliki enam belas juta warna dirubah menjadi 256 warna. Citra-citra keluaran yang dihasilkan adalah :



Gambar 5.3. Citra-citra keluaran pada percobaan kedua.

Percobaan ketiga dilakukan dengan melakukan perubahan resolusi sebesar sebelas persen terhadap citra masukan. Citra mula-mula berolusi 128x128 menjadi 144x144 . Citra-citra keluaran yang dihasilkan adalah :



Gambar 5.4. Citra-citra keluaran pada percobaan ketiga

Sedangkan untuk perubahan resolusi tiga belas persen (147×147) menyebabkan penurunan peringkat dari citra target pencarian hingga peringkat lima belas pada urutan citra-citra hasil pencarian dan perubahan yang melebihi tiga belas persen menyebabkan hasil pencarian yang gagal.

Percobaan keempat dilakukan dengan menggunakan citra yang telah dikaburkan warnanya. Citra masukan yang digunakan adalah :



Gambar 5.5. Citra masukan pada percobaan keempat

Setelah dilakukan pencarian, citra-citra keluaran yang dihasilkan adalah :



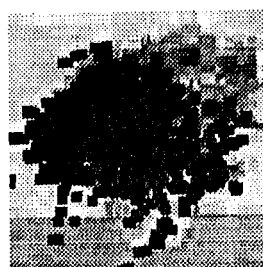
Gambar 5.6. Citra-citra keluaran pada percobaan keempat

Percobaan kelima dilakukan dengan menggunakan citra masukan yang telah mengalami konversi warna menjadi citra berskala abu-abu. Citra-citra keluaran yang dihasilkan adalah :



Gambar 5.7. Citra-citra keluaran pada percobaan kelima

Percobaan keenam dilakukan dengan menggunakan citra masukan yang telah mengalami pengrusakan secara sengaja dengan mengurangi sebagian informasi yang dimilikinya. Citra masukan yang digunakan adalah :



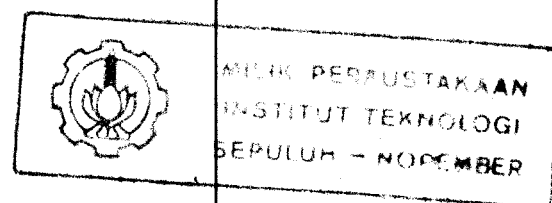
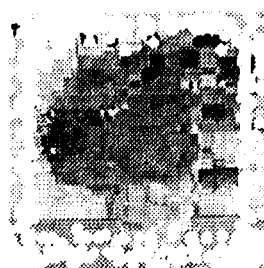
Gambar 5.8. Citra masukan percobaan keenam

Setelah dilakukan pencarian didapatkan citra-citra keluaran sebagai berikut :



Gambar 5.9. Citra-citra keluaran pada percobaan keenam

Percobaan ketujuh juga dilakukan dengan menggunakan citra masukan yang telah mengalami pengrusakan. Citra yang digunakan sebagai citra masukan adalah :



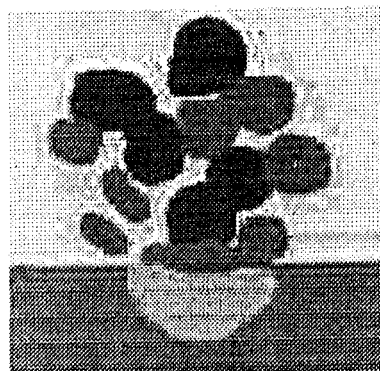
Gambar 5.10. Citra masukan percobaan ketujuh

Setelah dilakukan pencarian didapatkan citra-citra keluaran sebagai berikut :



Gambar 5.11. Citra-citra keluaran pada percobaan ketujuh

Percobaan kedelapan dilakukan dengan menggunakan citra sketsa sebagai citra masukan. Citra masukan yang digunakan adalah :



Gambar 5.12. Citra masukan pada percobaan kedelapan

Setelah dilakukan proses pencarian didapatkan citra-citra keluaran sebagai berikut:



Gambar 5.13. Citra-citra keluaran pada percobaan kedelapan

5.1.2 Analisis Pengaruh Perubahan Citra Terhadap Keberhasilan Pencarian

Setelah dilakukan percobaan dengan melakukan perubahan-perubahan terhadap kondisi citra masukan seperti pada bagian uji coba di atas, didapatkan data pengaruh perubahan sebagai berikut :

Tabel 5.1. Tabel data perubahan citra masukan terhadap keberhasilan pencarian.

Nomor	Perubahan	Nilai Beda	Koefisien Wavelet Sama (%)	Peringkat
1	Normal	-49,31998	100 %	1
2	Tingkat Warna	-48,47418	97,8 %	1
3	Resolusi (11%)	-31,35161	63,9 %	9
4	Pengaburan	-38,84983	78,9 %	1
5	Skala Abu-abu	-34,92626	70,6 %	6
6	Pengrusakan I	-41,99364	85 %	1
7	Pengrusakan II	-40,42517	80,6	18
8	Sketsa	-33,19081	63,9 %	9

Dari percobaan di atas dapat diketahui pengaruh-pengaruh yang ditimbulkan perubahan-perubahan tersebut terhadap keberhasilan pencarian, diantaranya:

1. Pengaruh perubahan tingkat warna tidak akan mempengaruhi keberhasilan pencarian. Perubahan ini tidak terlalu mempengaruhi nilai rerata warna dan karakteristik penyebaran koefisien wavelet yang dihasilkan. Hal tersebut disebabkan nilai warna citra yang didekomposisi adalah nilai warna yang telah dikonversikan ke dalam skala $[0,1]$. Sehingga tingkat warna menjadi tidak berpengaruh lagi.
2. Perubahan skala resolusi citra sangat mempengaruhi keberhasilan pencarian. Perubahan skala sangat mempengaruhi karakteristik penyebaran koefisien-koefisien wavelet dan rerata warna karena perubahan ukuran citra. Perubahan karakteristik penyebaran koefisien-koefisien wavelet dan rerata warna ini akan mempengaruhi nilai beda (*score*) hasil perbandingan. Pada uji coba

didapatkan toleransi perubahan skala dari citra masukan adalah hingga tiga belas persen (13%), namun hal ini menyebabkan penurunan peringkat citra target pada urutan citra-citra hasil pencarian.

3. Perubahan citra masukan menjadi citra dengan skala warna abu-abu mempengaruhi peringkat citra target pada urutan citra-citra hasil pencarian. Hal ini disebabkan perubahan nilai-nilai struktur warna R,G,B pada setiap piksel citra mempengaruhi karakteristik penyebaran koefisien-koefisien wavelet dan rerata warna citra karena dekomposisi dilakukan pada setiap nilai struktur warna R,G,B pada setiap piksel citra. Pada ukuran basisdata yang besar hal ini dapat menyebabkan proses pencarian mengalami kegagalan.
4. Pengaburan warna citra tidak mempengaruhi keberhasilan pencarian karena pengaburan terjadi pada seluruh bagian citra sehingga mempengaruhi nilai beda yang dihasilkan oleh seluruh pustaka citra secara merata. Pengaburan banyak mempengaruhi karakteristik penyebaran nilai-nilai koefisien wavelet namun tidak terlalu mempengaruhi nilai rerata warna citra karena pengaburan terjadi pada seluruh bagian citra. Perbedaan nilai rerata warna memiliki nilai faktor bobot yang besar pada penentuan nilai beda dan pengaburan warna tidak terlalu mempengaruhi rerata warna dari citra.
5. Kerusakan citra atau hilangnya informasi pada sebagian citra tidak akan mempengaruhi keberhasilan pencarian jika kerusakan tersebut tidak terlalu mempengaruhi nilai rerata warna dan karakteristik penyebaran koefisien-koefisien waveletnya. Pengaruh perubahan nilai rerata warna lebih berpengaruh terhadap nilai beda dari pada perubahan karakteristik penyebaran

koefisien-koefisien waveletnya sesuai dengan persamaan metrik (3.4), sehingga pengaruh kerusakan citra terhadap keberhasilan akan sangat bergantung pada pengaruhnya terhadap rerata warna. Kerusakan kecil pada citra biasanya tidak terlalu berpengaruh terhadap nilai rerata dan karakteristik penyebaran koefisien-koefisien wavelet. Pengaruh yang ditimbulkan kerusakan citra bisa berupa penurunan peringkat keberhasilan hingga gagalnya proses pencarian.

6. Pada penggunaan citra masukan yang berupa sketsa, keberhasilan pencarian akan sangat tergantung pada sketsa yang digunakan. Apabila sketsa yang digunakan memiliki banyak kesamaan dengan pustaka citra yang diinginkan, meliputi bentuk gambar, warna serta ukurannya akan memberikan nilai rerata warna dan karakteristik penyebaran nilai-nilai koefisien-koefisien wavelet yang hampir sama. Hal tersebut memungkinkan proses pencarian akan berhasil walaupun pada tingkat keberhasilan yang beragam. Proses pencarian dengan masukan berupa sketsa, keberhasilannya sangat tergantung kepada keadaan sketsa yang digunakan.

5.1.3 Uji Coba Untuk Mengetahui Tingkat Keberhasilan Pencarian

Uji coba ini dilakukan untuk mengetahui keberhasilan pencarian dari perangkat lunak yang dibuat. Uji coba dilakukan dengan berulang-ulang dengan menggunakan dua kriteria citra masukan, yakni :

- Citra masukan berupa citra masukan hasil *scan* yang juga terdapat pada basisdata. Percobaan dengan citra masukan jenis ini dilakukan sebanyak

seratus kali dengan citra masukan yang berbeda-beda. Proses pencarian yang dilakukan selalu berhasil dan pustaka citra yang diinginkan selalu berada pada peringat pertama. Keberhasilan dari uji coba dengan citra masukan jenis ini adalah seratus persen (100%).

- Citra masukan yang digunakan adalah citra yang telah mengalami penurunan kualitas warna (pengaburan) dan citra masukan yang berupa sketsa. Proses uji coba dilakukan sebanyak dua puluh kali dengan tingkat keberhasilan sembilan puluh persen (90%). Dari dua puluh percobaan terdapat dua kegagalan. Kegagalan terjadi pada uji coba dengan citra masukan berupa sketsa. Hal ini dimungkinkan karena terlalu banyaknya perbedaan antara citra sketsa dengan citra asli yang dimaksudkan.

5.2 ANALISIS KECEPATAN PENCARIAN

Percobaan pencarian dilakukan terhadap basisdata yang berisi 120 citra dan dilakukan dengan menggunakan PC 486 DX-2 , RAM 16 M dengan sistem operasi Wndows-95. Ternyata didapatkan bahwa pencarian dilakukan dengan sangat cepat karena proses pencarian hanya dilakukan dengan membandingkan koefisien-koefisien wavelet terpilih untuk setiap struktur warna R,G,B bukan dengan membandingkan secara langsung setiap citra target yang ada pada basisdata dengan citra masukannya. Proses tersebut hanya membutuhkan waktu satu detik untuk perbandingan guna mendapatkan citra-citra yang dianggap bersesuaian dengan citra masukan yang diberikan. Permasalahan waktu justru terjadi pada bagian praproses yakni proses dekomposisi wavelet terhadap citra-

citra pada basisdata dan citra masukan yang digunakan. Waktu proses yang dibutuhkan sangatlah bergantung terhadap resolusi dari citra yang didekomposisi. Hal ini terjadi karena proses dekomposisi dilakukan terhadap setiap nilai dari setiap struktur warna R,G,B pada setiap piksel yang terdapat pada citra. Hasil dari uji coba tentang perhitungan waktu dekomposisi dan pencarian dengan beragam ukuran dari citra masukan ditunjukkan pada tabel 5.1 di bawah ini.

Tabel 5.2. Tabel kecepatan proses dekomposisi dan pencarian citra berdasarkan ukuran citra masukan.

Nomor	Nama File Masukan	Ukuran Citra	Lama Proses Dekomposisi (detik)	Lama Pencarian (detik)	Total waktu (deetik)
1	Setir.bmp	64x64	0.5	1	1.5
2	Bunga.bmp	128x128	2	1	3
3	Handshak.bmp	240x180	5	1	6
4	Postman.bmp	293 x 429	17	1	18
5	Plus!.bmp	1024 x 768	170	1	171

Tampak pada data tabel di atas bahwa waktu dekomposisi yang dibutuhkan meningkat tajam pada ukuran citra yang besar. Hal tersebut disebabkan karena proses dekomposisi memang dilakukan terhadap keseluruhan piksel yang terdapat pada citra. Sehingga lama proses dekomposisi sangat tergantung pada ukuran dari citra yang didekomposisi. Waktu proses perbandingan tidak tergantung pada ukuran dari citra karena setiap citra diwakili oleh jumlah koefisien wavelet yang sama dan tidak tergantung pada ukurannya. Hal tersebut menyebabkan waktu proses perbandingan yang sama untuk ukuran citra masukan yang berbeda-beda.

BAB VI

PENUTUP

Pada bab ini akan diberikan kesimpulan dan saran sebagai hasil akhir dari penelitian tugas akhir yang dilakukan.

6.1 Kesimpulan

Dari hasil pengujian terhadap perangkat lunak yang dibuat maka dapat disimpulkan hal-hal sebagai berikut :

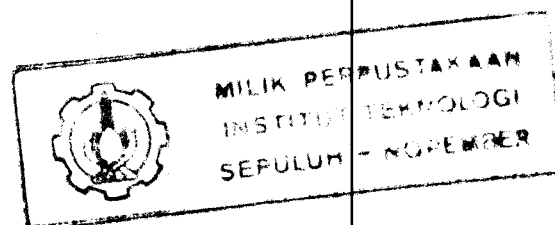
1. Sistem pencarian citra terhadap basisdata citra dengan metode dekomposisi wavelet dapat melakukan pencarian dengan sangat cepat dan akurat bahkan terhadap basisdata berukuran besar sekalipun karena memang proses pencarian tidak dilakukan dengan melakukan perbandingan langsung citra-citra pada basisdata dengan citra masukan tetapi dilakukan dengan membandingkan koefisien-koefisien wavelet terpilih hasil dekomposisi terhadap citra.
2. Waktu proses (kecepatan) dari pencarian tidak tergantung pada ukuran citra masukan, karena setiap citra diwakili oleh jumlah koefisien yang sama tak tergantung pada ukurannya. Waktu terbesar yang dibutuhkan adalah waktu yang diperlukan untuk melakukan dekomposisi citra masukan. Waktu Dekomposisi tergantung pada ukuran resolusi dari citra yang didekomposisi. Semakin besar ukuran citra atau semakin banyak piksel yang terdapat pada suatu citra membutuhkan waktu dekomposisi yang lebih lama dengan

peningkatan waktu yang sangat ekstrim. Waktu dekomposisi dianggap sebagai praproses.

3. Metode pencarian citra terhadap basisdata citra dengan metode dekomposisi wavelet dapat digunakan sebagai metode pencarian alternatif yang baik dengan ketepatan pencarian yang tinggi. Terutama jika citra-citra yang terlibat pada proses pencarian telah distandarisasi ukurannya.

6.2 Saran

Dengan melihat pada hasil yang ada, maka untuk pengembangan lebih lanjut dapat dikembangkan proses pencarian terhadap basisdata dengan citra-citra bergerak (Video Querying). Selain itu kelemahan dari pencarian citra dengan metode dekomposisi wavelet, yakni lamanya waktu dekomposisi yang dibutuhkan dapat lebih dipersingkat dengan algoritma pemrograman *parallel processing*. Meskipun pada kenyataannya proses dekomposisi tidak terlibat langsung pada proses pencarian, namun dengan waktu dekomposisi yang singkat maka praproses yang merupakan proses persiapan dari basisdata untuk menghasilkan koefisien-koefisien yang akan digunakan untuk mewakili basisdata pada proses pencarian dapat dilakukan dengan lebih cepat terutama untuk basisdata dengan ukuran besar.



DAFTAR PUSTAKA

1. Charles E. Jacobs, Adam Finkelstein, and David H. Salesin [1995], "Fast Multiresolution Image Querying", In Proceeding of SIGGRAPH 95, ACM, New York.
2. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin [1995], "Wavelet for Computer Graphics: Primer Part I", IEEE Computer Graphics and Application, May, pp. 76-84.
3. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin [1995], "Wavelet for Computer Graphics: Primer Part II", IEEE Computer Graphics and Application, May, pp. 76-84.
4. Oscar Mangisengi [1996], "Deteksi tepi dalam citra dengan metode wavelet", M.K. Tesis, Universitas Indonesia, Jakarta .
5. Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin [1995], "Wavelet for computer graphics : Theory and Applications", Morgan Kaufmann, San Fransisco.
6. Howard Anton, Pantur Silaban [1991], "Aljabar linier Elementer", edisi ketiga, Pernerbit Erlangga, Jakarta.
7. David Meyers [1994], "Multiresolution tiling" ,Computer Graphics Forum, 13(5):325-340.